THE UNIVERSITY OF CHICAGO

MAKING INTERACTIONS WITH HOME IOT DEVICES MORE SECURE, PRIVATE, AND USABLE

A THESIS PROPOSAL SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCE
IN CANDIDACY FOR THE DEGREE OF
PHILOSOPHY OF DOCTOR

DEPARTMENT OF COMPUTER SCIENCE

BY
WEIJIA HE

CHICAGO, ILLINOIS

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The internet of things for homes (home IoT) brings unique challenges to security. These home IoT devices often interact with multiple people under the same roof and are equipped with various modalities. They don't only react to commands from the user but also from the environment, which increases the attacking surface and changes the threat model. Home IoT devices' highly fragmented ecosystem only makes things worse, making it harder to find a solution that fits all.

Traditional security measurements fail in these challenges because they are designed for conventional computing devices like computers or smartphones, which are mostly used by one user with proper screens and keyboards. These premises make mechanisms like access control and authentication much more manageable. On the other hand, traditional computing devices are all general-purposed, making enforcing allowlists impossible. This is no longer the case for home IoT devices, and new strategies must be taken.

Responding to these emerging challenges in home IoT, we create a road map about how to make a home IoT system secure and usable on different levels. We are mainly interested in the device's interactions with the external world, such as users, environments, and remote servers. With such emphasis, we divide a home IoT system into three parts: user & software, environment & hardware, and network. For the user & software part, we survey what an access control system needs for complicatedly associated users and constantly changing contexts. For environments & hardware, we create a framework for context sensing, systematizing contexts and their required sensors, along with the security, privacy, and usability promises they hold. In the network part, we explore the design space of creating an allowlist that can work for various devices of one kind.

CHAPTER 1

INTRODUCTION

The proliferation of home IoT devices in recent years has raised significant security and privacy concerns [1]. Unlike traditional computing devices such as computers or smartphones, home IoT devices often need to face multiple users with complex social relationships and react to multiple modalities [2, 3]. Both create a larger attack surface than before. Moreover, many IoT devices are not designed for general purpose. Devices with various purposes inherently work in different ways. Even for devices made for the same purpose, different manufactures create different network architecture, which causes more fragmentation in the home IoT ecosystem [4]. The widely existing variations in home IoT devices do not only cause security flaws everywhere but also make security measurements hard to protect all the devices in the wild [5, 6, 7].

Unfortunately, many security practices and mindsets fail to recognize these changes and address them properly. Access control in smart homes, for example, retains the admin-guests model that we used to have on computers, despite the fact that the social relationships between users are much more complicated [8, 9, 10, 11, 12, 13, 14]. IoT systems are also sensitive to environmental changes, attackers can alter the system's behaviors by changing the environment [15, 16, 17, 18, 19, 20]. Security and privacy research for home IoT devices is often dedicated only to one type of devices [21, 22], or some particular home IoT platforms [23, 24]. Even for research claims to be widely applicable, it is hardly evaluated on a large scale.

In this proposal, we revisit and inspect various aspects of a home IoT system, rethink the old mindsets and how they fail in face of the unique challenges brought by home IoT, and propose more adaptive, secure, and privacy-respectful designs. The proposal emphasizes on the interactions a home IoT device may have with the external world, instead of the internal built of the device. To systematically examine a home IoT system, we divide the external

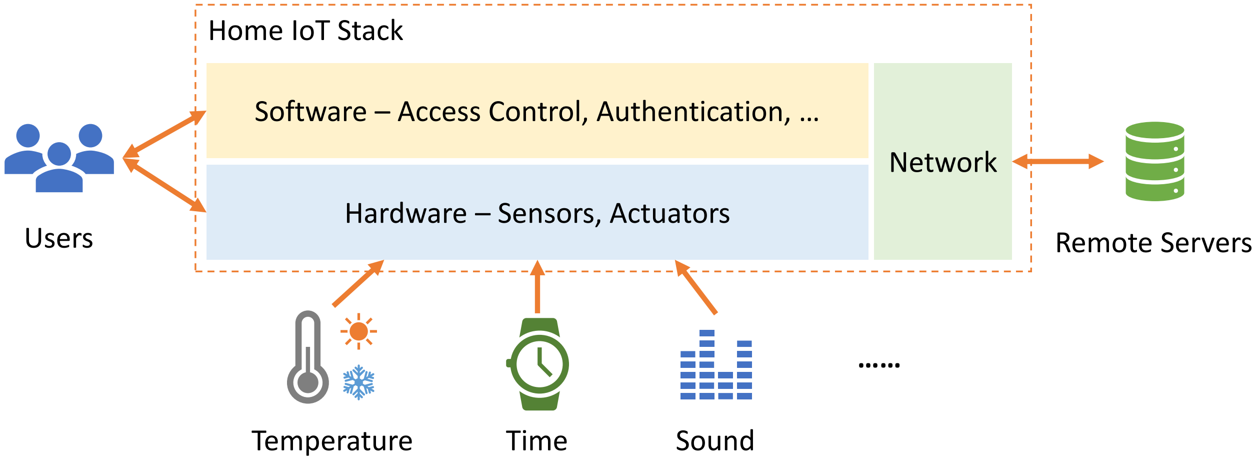

Home IoT Stack
Software – Access Control, Authentication, ...
Hardware – Sensors, Actuators
Network
Users
Remote Servers
Temperature
Time
Sound
......

Figure 1.1: The home IoT stack model we used in this proposal. It emphasize the interactions a home IoT may have with the external world, namely users, surroundings, and remote servers.

interactions into three different parts:

- **Users & Software:** This part mainly consists of most of the software, including the ones about security. This is also the part which users actively interact with. In this proposal, we are interested in access control and authentication, and their interactions with users. We studied pitfalls of these fundamental security mechanisms in the age of IoT, and explored how they should be built through large-scale user studies. We discovered that a desirable access control and authentication system should respect the delicacy of different social relationship among users, and be adaptive the the constantly changing contexts in a home [8].

- **Environment & Hardware:** All the home IoT has its own hardware deployment. The hardware supports the functionalities of the home IoT device, such as various sensors it uses to understand its surrounding and environmental changes. Based off the idea of context-aware access control, we explore various desired contexts and find assorted sensors than can actually sensed them. Moreover, we put them under an adversarial setting and create a framework that can evaluate their robustness to various attacks, along with their privacy-respectfulness and usability in the lifecycle [25].

- **Network:** The network part here is all about how home IoT devices communicate with remote servers. Firewalls are the main topic in security on this layer. Our ongoing work explores the design space of automatically creating generalizable allowlists for various home IoT products. We attempts to create allowlists that can work for all devices of a product, instead of just one device, by using the IoT Inspector dataset, a large-scale IoT traffic dataset form the wild [26].

Throughout the proposal, we imagine to create a road map that can secure home IoT in different stages from various attackers. A smart home owner could first use our framework about context sensing to select a combination of home IoT devices that not only fit their needs, but also be robust against local attackers and be privacy-preserving as well. Once the selected home IoT devices are deployed, the owner can use allowlists to prevent attacks launched by a remote attacker or a compromised device. Finally, for legitimate users, a good access control system will assign default access based on their social roles in the household, as suggested by our study.

CHAPTER 2

RETHINKING ACCESS CONTROL FOR THE HOME IOT

Recent years have seen a proliferation of Internet of Things (IoT) devices intended for consumers' homes, including Samsung SmartThings [27], the Amazon Echo voice assistant [28], the Nest Thermostat [29], Belkin's Wemo devices [30], and Philips Hue lights [31]. To date, IoT security and privacy research has focused on such devices' insecure software-engineering practices [32, 33, 34], improper information flows [33, 23, 35], and the inherent difficulties of patching networked devices [5, 6].

Surprisingly little attention has been paid to *access-control-policy specification* (expressing which particular users, in which contexts, are permitted to access a resource) or *authentication* (verifying that users are who they claim to be) in the home IoT. This state of affairs is troubling because the characteristics that make the IoT distinct from prior computing domains necessitate a rethinking of access control and authentication. Traditional devices like computers, phones, tablets, and smart watches are generally used by only a single person. Therefore, once a user authenticates to their own device, minimal further access control is needed. These devices have screens and keyboards, so the process of authentication often involves passwords, PINs, fingerprint biometrics, or similar approaches [36].

Home IoT devices are fundamentally different. First, numerous users interact with a single home IoT device, such as a household's shared voice assistant or Internet-connected door lock. Widely deployed techniques for specifying access-control policies and authenticating users fall short when multiple users share a device [2]. Complicating matters, users in a household often have complex social relationships with each other, changing the threat model. For example, mischievous children [9], parents curious about what their teenagers are doing [10], and abusive romantic partners [37] are all localized threats amplified in home IoT environments.

Furthermore, few IoT devices have screens or keyboards [3], so users cannot just type a password. While users could possibly use their phone as a central authentication mechanism, this would lose IoT devices' hands-free convenience, while naïve solutions like speaking a password to a voice assistant are often insecure.

Real-world examples of the shortcomings of current access-control-policy specification and authentication for home IoT devices have begun to appear. A Burger King TV commercial triggered Google Home voice assistants to read Wikipedia pages about the Whopper [17], while the cartoon South Park mischievously triggered Amazon Echo voice assistants to fill viewers' Amazon shopping carts with risqué items [16]. While these examples were relatively harmless, one could imagine a rogue child remotely controlling the devices in a sibling's room to annoy them, a curious babysitter with temporary access to a home perusing a device's history of interactions, or an enterprising burglar asking a voice assistant through a cracked window to unlock the front door [15].

In this paper, we take a first step toward rethinking the specification of access-control policies and authentication for the home IoT. We structure our investigation around four research questions, which we examine in a 425-participant user study. These research questions are motivated by our observation that many home IoT devices combine varied functionality in a single device. For example, a home hub or a voice assistant can perform tasks ranging from turning on the lights to controlling the door locks. Current access control and authentication is often based on a device-centric model where access is granted or denied per device. We move to a capability-centric model, where we define a capability as a particular action (e. g., ordering an item online) that can be performed on a particular device (e. g., a voice assistant). Intuition suggests that different capabilities have different sensitivities, leading to our first research question:

RQ1: Do desired access-control policies differ among capabilities of single home IoT devices? (Section 2.4.2 and 2.4.3).

We investigated this question by having each study participant specify their desired access-control policy for one of 22 home IoT capabilities we identified. For household members of six different relationships (e. g., spouse, child, babysitter), the participant specified when that person should be allowed to use that capability. Our findings validated our intuition that policies about capabilities, rather than devices, better capture users' preferences. Different capabilities for voice assistants and doors particularly elicited strikingly different policies.

While the ability to specify granularly who should be able to use which capabilities is necessary to capture users' policies, it incurs a steep usability cost. To minimize this burden through default policies, we asked:

RQ2: For which pairs of relationships (e. g., child) and capabilities (e. g., turn on lights) are desired access-control policies consistent across participants? These can be default settings (Section 2.4.4).

In our study, nearly all participants always wanted their spouses to be able to use capabilities other than log deletion at all times. Participants also wanted others to be able to control the lights and thermostat while at home. As intimated by the prior policy, the context in which a particular individual would use a capability may also matter. Children might be permitted to control lights, but perhaps not to turn the lights on and off hundreds of times in succession as children are wont to do. Nor should children be permitted to operate most household devices when they are away from home, particularly devices in siblings' rooms. A babysitter unlocking the door from inside the house has far fewer security implications than the babysitter setting a persistent rule to unlock the front door whenever anyone rings the doorbell.

RQ3: On what *contextual factors* (e. g., location) do access-control policies depend? (Section 2.4.5).

In addition to a user's location, we found that participants wanted to specify access-control policies based on a user's age, the location of a device, and other factors. Almost none of these contextual factors are supported by current devices. Finally, to identify promising directions for designing authentication mechanisms in the home IoT, we asked:

RQ4: What types of authentication methods balance convenience and security, holding the potential to successfully balance the consequences of falsely allowing and denying access? (Section 2.4.6).

Analyzing consequences participants noted for falsely allowing or denying access to capabilities, we identify a spectrum of methods that seem promising for authenticating users (Section 2.5), thereby enabling enforcement of users' desired access-control policies for the home IoT.

2.1 Background

In this section, we scope our notion of home IoT devices, identify our threat model, and review current devices' support for access control and authentication.

We define home IoT devices to be small appliances that are Internet-connected and used primarily in the home. Internet-connected lights and thermostats are two examples. Many such devices are managed through a hub that facilitates communication between devices, enforces policies, and often allows for the creation of end-user programs or the use of apps.

2.1.1 Threat Model

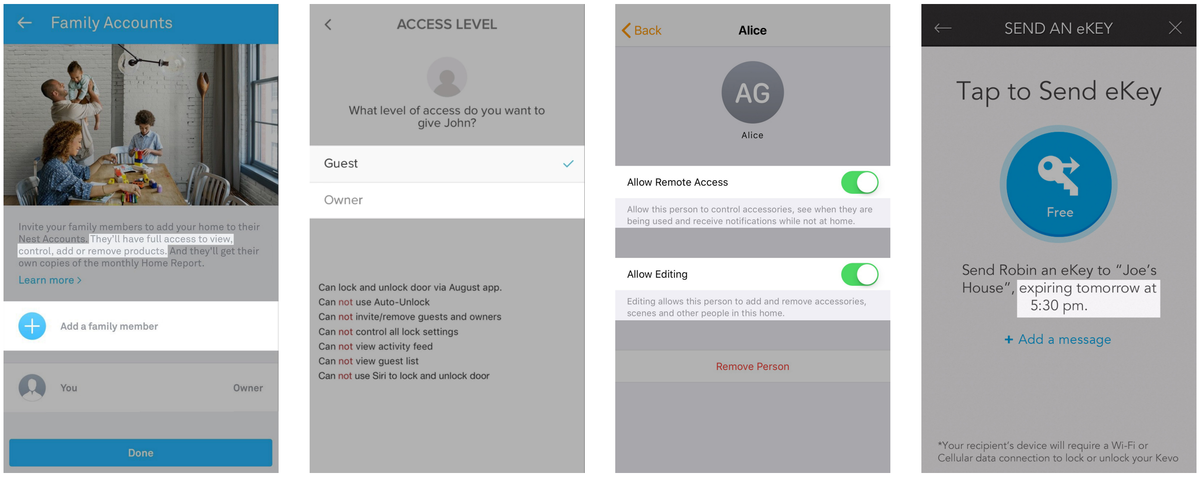
The two major classes of adversaries in the smart home are external third parties and those who have legitimate physical access to the home. The former class includes those who exploit software vulnerabilities in platforms [34], devices [32] (e. g., with Mirai), or protocols [38] intending to cause physical, financial, or privacy-related damage. The latter class includes household members with legitimate digital or physical access to the home, such as temporary workers or children [9]. These insider threats have received far less research attention, but are the focus of this paper. Insiders might be motivated to subvert a smart-home system's access controls for reasons ranging from curiosity to willful disobedience (e.g., a child attempting to take actions forbidden by their parents), or to attempt to correct imbalances created by the introduction of devices whose surveillance implications grant asymmetric power to certain members of a household (e. g., a parent tracking a teenager [10]).

We assume a domestic setting where occupants control home IoT devices through smartphones, voice assistants, rules, and physical interaction. For example, a maintenance worker may unlock the front door using a smartphone app, while a child might turn off their lights by speaking to a voice assistant. We aim for access-control rules that balance security, privacy, and functionality.

2.1.2 Affordances of Current Devices

Current home IoT devices have relatively limited affordances for access control and authentication. Taking a five-year-old survey of the home IoT landscape as a starting point [39], we surveyed current devices' affordances; Figure 2.1 shows representative samples. To control many current devices, people use smartphone apps that must be paired with devices. These apps offer various access-control settings. For example, the Nest Thermostat supports a binary model where additional users either have full or no access to all of the thermostat's capabilities. The August Smart Lock offers a similar model with guest and owner levels. Withings wireless scales let users create separate accounts and thus isolate their weight measurements from other users. On Apple HomeKit, one can invite additional users, restricting them to: (a) full control, (b) view-only control, (c) local or remote control.

Some devices offer slightly richer access-control-policy specification. The Kwikset Kevo Smart Lock allows access-control rules to be time-based; an owner can grant access to a secondary user for a limited amount of time. We find in our user study that time is a desirable contextual factor, but one of only many. We focus on capabilities, rather than devices. While most current devices do not allow for access-control policies that distinguish by capability, Samsung SmartThings lets users restrict third-party apps from accessing certain capabili-



Invite your family members to add your home to their Nest Accounts. They'll have full access to view, control, add or remove products. And they'll get their own copies of the monthly Home Report.

What level of access do you want to give John?

Can lock and unlock door via August app.
Can not use Auto-Unlock
Can not invite/remove guests and owners
Can not control all lock settings
Can not view activity feed
Can not view guest list
Can not use Siri to lock and unlock door

Allow Remote Access: on

Allow this person to control accessories, see when they are being used and receive notifications while not at home.

Allow Editing: on

Editing allows this person to add and remove accessories, scenes and other people in this home.

Send Robin an eKey to "Joe's House", expiring tomorrow at 5:30 pm.

*Your recipient's device will require a Wi-Fi or Cellular data connection to lock or unlock your Kevo

(a) Nest Learning Thermostat (b) August Smart Lock (c) Apple HomeKit (d) Kwikset Kevo Smart Lock

Figure 2.1: Current access-control-specification interfaces: The Nest Thermostat (a) only allows "all-or-nothing" specification, while the August Smart Lock (b) only offers coarse-grained access control via predefined Guest and Owner groups. In contrast, Apple's HomeKit (c) differentiates between view and edit access level, as well as local and remote access. The Kwikset Kevo Smart Lock (d) provides time-based access control, but not other factors.

ties [40]. We find that restricting users, not just apps, access to a particular capability is necessary.

From this analysis, we found current mechanisms to be rudimentary and lack the necessary vocabulary for specifying access-control rules in complex, multi-user environments. We aim to establish a richer vocabulary.

Current authentication methods for the home IoT appear transplanted from smartphone and desktop paradigms. Passwords are widely used in conjunction with smartphones. For example, SmartThings has an app through which a user can control devices. A user first authenticates to this app using a password. Voice-based authentication is currently very rudimentary and is not used for security, but for personalization. For instance, Google Home uses speaker recognition for customizing reminders, but not for security-related tasks [41].

2.2 Pre-Study

As a first step in exploring access control based on capabilities and relationships in the home IoT, we conducted a pre-study to identify capabilities and relationships that elicit representative or important user concerns. To ground our investigation of capabilities of the home IoT in devices consumers would likely encounter, we created a list of home IoT devices from consumer recommendations in CNET, PCMag, and Tom's Guide [42]. We grouped devices by their core functionality into categories including *smart-home hubs*, *door locks*, and *voice assistants*.

For each category of device, we collected the capabilities offered by currently marketed devices in that category. We added likely future capabilities, as well as the ability to write end-user programs [23, 35]. We showed each pre-study participant all capabilities identified for a single given class of device. The participant answered questions about the positive and negative consequences of using that capability, and they also identified additional capabilities they expected the device to have. We used this process to identify a comprehensive, yet diverse, set of capabilities that range from those that elicit substantial concerns to those that elicit none.

To identify a small set of relationships to investigate in the main study, we also showed participants a table of 24 relationships (e. g., teenage child, home health aide) and asked them to group these relationships into five ordered levels of desired access to smart-home devices. We chose this list of 24 relationships based on existing users and groups in discretionary access control (DAC) systems and common social relationships in households.

We conducted the pre-study with 31 participants on Amazon's Mechanical Turk. Participants identified potential concerns for a number of capabilities, in addition to identifying capabilities (e. g., turning on lights) that aroused few concerns. We used these results to generate a list of capabilities, grouping similar functionalities across devices into categories like viewing the current state of a device. We selected the 22 capabilities whose pre-study results showed a spectrum of opinions and concerns while maintaining a feature-set representative of smart homes.

To narrow our initial list of 24 relationships to a tractable number, we examined how pre-study participants assigned each relationship to one of the five ordered categories of desired access to household devices. We chose the six relationships that span the full range of desired access and for which participants were most consistent in their assignments to a category.

2.3 Main Study

To elicit desired access-control policies for the home IoT, our main study was an online survey-based user study. We recruited participants on Mechanical Turk, limiting the study to workers age 18+ who live in the United States and have an approval rating of at least 95 %.

2.3.1 Protocol

Each participant was presented with a single capability (e.g., "see which lights in the home are on or off") randomly chosen from among the 22 identified in the pre-study.

We then presented the participant with one of six relationships: spouse; teenage child; child in elementary school; visiting family member; babysitter; neighbor. We first asked whether such a person should be permitted to control that capability "always," "never," or "sometimes, depending on specific factors." These answers were the first step in identifying participants' desired access-control policies. For the first two options, we required a short free-text justification. To better understand the importance of an authentication method correctly identifying the person in question and the system correctly enforcing the access-control policy, we asked participants who answered "always" or "never" to state how much of an inconvenience it would be if the system incorrectly denied or allowed (respectively) that particular user access to that capability. Participants chose from “not an inconvenience,” “minor inconvenience,” or “major inconvenience,” with a brief free-text justification.

If the participant chose “sometimes,” we required additional explanations to further delineate their desired access-control policy. They first explained in free-text when that person should be allowed to use that capability, followed by when they should not be allowed to do so. On a five-point scale from “not important” to “extremely important,” we asked how important it was for them to have (or not have) access to that capability.

We repeated these questions for the other five relationships in random order. Thus, each participant responded for all six relationships about a single capability.

Afterwards, we asked more general questions about specifying access-control policies for that capability. In particular, we presented eight contextual factors in randomized order, asking whether that factor should influence whether or not anyone should be permitted to use that capability. The possible responses were “yes,” “no,” and “not applicable,” followed by a free-response justification. We asked about the following factors: the time of day; the location of the person relative to the device (e.g., in the same room); the age of the person; who else is currently at home; the cost of performing that action (e.g., cost of electricity or other monetary costs); the current state of the device; the location of the device in the home; the person’s recent usage of the device. Further, we asked participants to list any additional factors that might affect their decision for that capability.

We concluded with questions about demographics, as well as the characteristics of the participant’s physical house and members of their household. We also asked about their ownership and prior use of Internet-connected devices. We compensated participants $ 3.50 for the study, which took approximately 20 minutes and was IRB-approved.

2.3.2 Analysis

Participants' responses about their access-control preferences included both qualitative free-text responses and multiple-choice responses. Two independent researchers coded the qualitative data. The first researcher performed open coding to develop a code book capturing the main themes, while the second coder independently used that same code book. To quantitatively compare multiple-choice responses across groups, we used the chi-squared test when all cell values were at least 5, and Fisher's Exact Test (FET) otherwise. For all tests, $\alpha = .05$, and we adjusted for multiple testing within each family of tests using Holm correction.

2.3.3 Limitations

The ecological validity and generalizability of our study are limited due to our convenience sample on Mechanical Turk. Most of our questions are based on hypothetical situations in which participants imagine the relationships and capabilities we proposed to them and self-report how they expect to react. Furthermore, while some participants were active users of home IoT devices, others were not, making the scenarios fully hypothetical for some participants. We chose to accept this limitation and include recruits regardless of prior experience with home IoT devices to avoid biasing the sample toward early adopters, who tend to be more affluent and tech-savvy.

2.4 Results

In the following sections we present our findings. We begin by providing an overview of our participants (Section 2.4.1). Next, we present how desired access-control policies differ across capabilities (RQ1, Section 2.4.2) and the degree to which desired policies differ across relationships (RQ1, Section 2.4.3). After that, we show for which pairs of relationships and capabilities the desired access-control policies are consistent across participants. We use these pairs to derive default policies (RQ2, Section 2.4.4). Next, we evaluate which contextual factors (e. g., age, location, usage) influence the “sometimes” cases the most, thus explaining users’ reasoning for not always allowing access to a capability (RQ3, Section 2.4.5). Finally, we analyze the consequences of false authorization and show the impact of falsely allowing / denying access to a certain capability on a per-relationship level (RQ4, Section 2.4.6).

2.4.1 Participants

A total of 426 individuals participated in the study, and 425 of them were qualified as effective responses. One response was excluded from our data because their free-text responses were unrelated to our questions. Our sample was nearly gender-balanced; 46 % of participants identified as female, and 54 % as male. The median age range was 25-34 years old (47 %). Most participants (85 %) were between 25 and 54 years old. Some participants (19 %) reported majoring, earning a degree, or holding a job in computer science or a related field.

The majority of our participants (67 %) live in a single-family home, while 25 % live in an apartment. Nearly half of the participants own (49 %) the place where they live, while 47 % rent. Furthermore, we asked how many people (including the participant) live in the same household. Around 20 % of participants reported living in a single-person household, 27 % in a two-person, 23 % in a three-person, and 17 % in a four-person household.

2.4.2 Capabilities (RQ1)

Current access-control implementation in a smart home system is largely device-based. However, our data motivates a more fine-grained, flexible access-control mechanism. In the following parts, we discuss our main findings, which are visualized in Figure 2.2.

A) **Capability Differences Within a Single Device**

We observed that participants’ attitudes toward various capabilities differ within a single

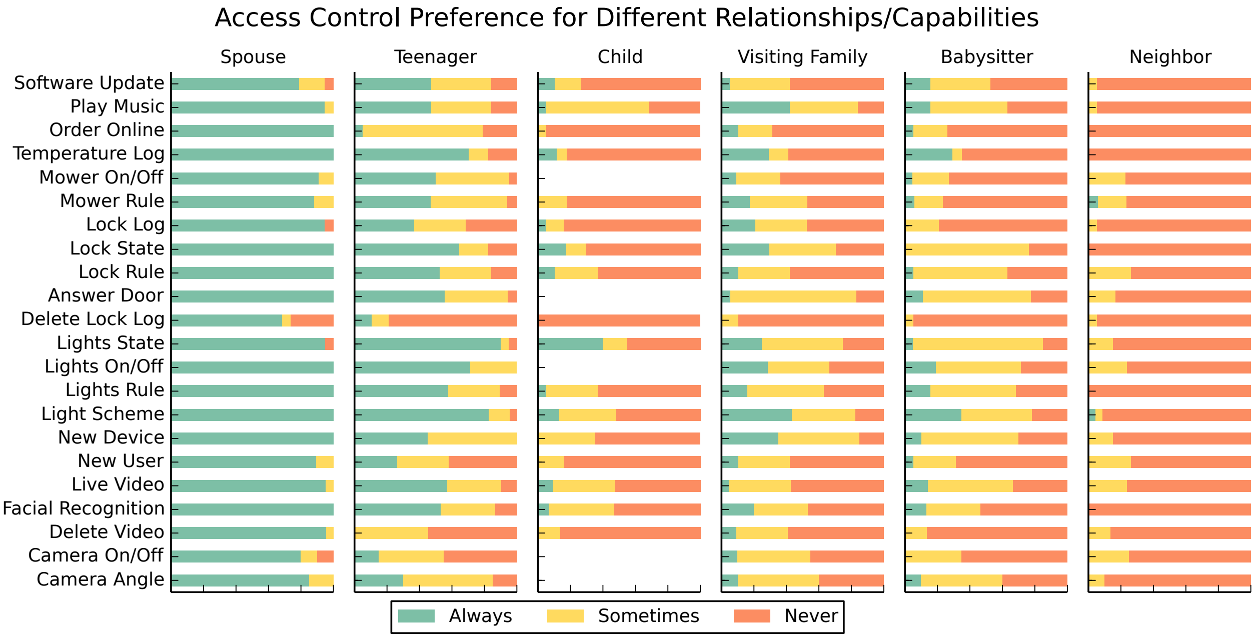


Figure 2.2: Participants' desired access-control policies. We introduced participants to a list of relationships (e.g., *neighbor*) and asked them to choose whether someone of that relationship should be permitted to "always," "sometimes," or "never" control a capability (e.g., adjust the *camera angle*) in their smart home.

device. For example, voice assistants can be used to play music and order things online. However, participants were much more willing to let others play music (32.5 % of participants choose *never* averaged across the six relationships, $\sigma = 0.33$, $median = 23.7\,\%$) than order things online (59.7 % choose *never* on average, $\sigma = 0.40$, $median = 71.1\,\%$) (FET, $p < .05$ for the teenager, child, and visiting family member relationships).

Another example of differing opinions across capabilities within a single device include deleting an IoT lock's activity logs and answering the door, viewing the current state of the lock, and setting rules for the lock. Across relationships, participants were permissive about capabilities like answering the door (25.6 % chose "never" averaged across all relationships other than children, $\sigma = 0.33$, $median = 16, 7\,\%$). Because children would likely not have a smartphone, we did not ask about them performing this action and we exclude them from this analysis. In contrast, 76.8 % of participants said they would *never* allow others to delete activity logs ($\sigma = 0.28$, $median = 92.1\,\%$). These differences are significant (FET, all $p < 0.05$ comparing within teenagers, visiting family, and babysitters). Even for a very trust-based relationship like a spouse, some participants still chose *never*. When asked why, one participant wrote: *"No one should be able to delete the security logs."*

Even if individuals with relationships like neighbor or babysitter do not live in the same house, permissions are sometimes given when the owner of the house is not around. One typical response for when a capability should be accessible to neighbors is *"Perhaps when I'm on vacation and I ask them to watch my home."*

B) **Context-Dependent Capabilities**

We identified "Answering the Doorbell" to be a highly context-dependent capability. 40 % of participants across relationships ($\sigma = 0.33$, $median = 38.9\,\%$) selected *sometimes* for this capability. At the same time, an average of 25.6 % of participants across relationships chose *never* ($\sigma = 0.33$, $median = 16.7\,\%$).

Whether the homeowner is present is a key factor impacting responses. Many participants (66.7 %) chose *sometimes* when it came to the babysitter, because the job itself indicates the parents are not around. If a delivery person rings the doorbell while the babysitter is home, the babysitter should be allowed to handle the event. The majority of participants (77.8 %) also *sometimes* trust a visiting family member with the same level of access. Some participants (16.7 %) will even consider giving this access to their neighbors, so that if there is an emergency when the family is on vacation, their neighbor can see who is at the door from their smartphone.

2.4.3 Relationships (RQ1)

Relationships play an important role in participants' preferred access-control policies.

A) **Babysitter vs. Visiting Family**

In the pre-study, we identified the babysitter and a visiting family member to be members of a guest-like group. In the main study, participants' overall attitudes toward babysitters and visiting family members were quite consistent with each other. No significant differences are observed between these two relationships in our pairwise chi-squared tests. This is understandable because both relationships share some trust with the homeowner, while neither lives in the same household.

In general, policies toward a visiting family member are slightly more permissive than policies toward a babysitter. However, analyzing the qualitative data, we found the situation to be more complex. There are some specific capabilities, such as "Live Video," where babysitters would be granted permissions at a higher rate than a visiting family member. 57.1 % of participants decided that a visiting family member would *never* have access to this feature, while only 33.3 % of participants decided the same for a babysitter. The reason is that a babysitter's job is to take care of a child while a parental figure is away. Therefore, the capability itself might help a babysitter take better care of the child, leading to a high rate of granting this permission *sometimes*.

Meanwhile, some features show strong subjective variations, including granting babysitters and visiting family members permission for "Answering the Doorbell." Some participants found it useful to always allow access, while other participants felt uncomfortable letting someone that is not part of their family have access to this particular capability.

From these observations, we conclude that it is important to have both a relationship-based and capability-based access-control model in a smart home. Such a model should be flexible enough to address the complex needs and use cases that might occur.

B) **Child vs. Teenager**

Though both children and teenagers are under a parent or guardian's watch, a teenager (presented as 16 years old) and a child (presented as 8 years old) were given very different access scopes. After removing the five capabilities that are not applicable to a child (whom we assume lacks a smartphone), for twelve of the seventeen remaining capabilities teenagers were given greater access (FET, all $p < .05$). A 16-year-old teenager was regarded as a young adult by many participants and was more widely trusted to use capabilities responsibly. Therefore, the *always* permission was chosen often, and no need for supervision was mentioned in their free-text responses.

Meanwhile, granting an 8-year-old child unencumbered access worried participants much more. Some participants mentioned that they were concerned that a young child would misuse these capabilities, either intentionally or unintentionally, and thus ruin all the settings. Several participants even expressed their worries that a young child could get themselves in danger with the access. For instance, one participant, who selected *never* for the capability of seeing which door is currently locked or unlocked, wrote: *"An elementary school child should not be leaving the house on his own accord."* An 8-year-old child's level of understanding of a smart home system is also questionable. As a result, children rarely were granted access *always* for capabilities other than those related to lights.

Even for capabilities for which participants chose relatively restrictive settings for both teenagers and young children (e. g., "Order Online"), attitudes differed. Though only 5.3 % of participants agreed to give full access to "Order Online" to a teenager, 73.7 % chose *sometimes* over *never*, giving limited access to their teenager to buy things they needed on Amazon. For young children, 94.7 % participants believed that a child at that age should *never* have access to it, frequently justifying that there is no need for younger children to order things online themselves. Many participants mentioned supervision or limitations on what a teenager can buy on Amazon, but they did admit they would let a teenager buy things from Amazon themselves if they had a reason.

C) **Overall Preference for Restrictive Polices**

We found that, except for spouses and teenagers, most participants preferred a more restrictive access-control policy over a more permissive one. For nine of the twenty-two capabilities averaged over all relationships, more than half of participants chose *never* more frequently

Table 2.1: Potential default access-control policies that reflected the vast majority of participants' preferences.

All

- *Anyone* who is *currently at home* should always be *allowed* to adjust *lighting*
- *No one* should be *allowed* to *delete log files*

Spouse

- *Spouses* should *always* have access to *all capabilities*, except for deleting log files
- *No one except a spouse* should unconditionally be allowed to access administrative features
- *No one except a spouse* should unconditionally be allowed to make online purchases

Children in elementary school

- Elementary-school-age *children* should *never* be able to use capabilities *without supervision*

Visitors (babysitters, neighbors, and visiting family)

- *Visitors* should only be able to use any capabilities *while in the house*
- *Visitors* should *never* be allowed to use capabilities of *locks, doors, and cameras*
- *Babysitters* should only be able to *adjust the lighting and temperature*

than *sometimes*, and *sometimes* more frequently than *always*. Averaged across all capabilities, only 18.1 % of participants ($\sigma = 0.12$, $\mathit{median} = 13.2\,\%$) chose *always* for visiting family members, 10.3 % for babysitters ($\sigma = 0.09$, $\mathit{median} = 7.9\,\%$), 8.3 % for children ($\sigma = 0.10$, $\mathit{median} = 5.6\,\%$) and 0.7 % for neighbors ($\sigma = 0.03$, $\mathit{median} = 0\,\%$). There was only a small group of capabilities for which participants were widely permissive: controlling lights and music, which do not have much potential to cause harm or damage.

2.4.4 Default Policies (RQ2)

In this section, we give an overview of the default deny/allow access policies we observed that capture most participants' responses. We categorize the policies by relationships and give an in-depth analysis of our findings.

Default Allow

A) **Spouses are Highly Trusted**

Averaged across all capabilities, 93.5 % of participants ($\sigma = 0.09$, $\mathit{median} = 95.3\,\%$) agreed to *always* give access to their spouse, while only 4.15 % ($\sigma = 0.05$, $\mathit{median} = 0\,\%$) answered *sometimes*, and 2.35 % ($\sigma = 0.06$, $median = 0\,\%$) said *never*. For participants who selected *always*, their most frequent reason was that they fully trust their spouse and that equality should be guaranteed in a marriage. Half of the non-permissive responses came from the capability to delete the smart lock's log file.

B) **Controlling Lights**

Access-control policies relating to lights were the most permissive. Looking at the responses for the capability to turn lights on and off, most responses align with a proposed default policy of people only being able to control the lights if they are physically present within the home. Relatedly, some participants chose *sometimes* for visiting family members and babysitters, depending on whether they are physically present within the home.

Default Deny

A) **Lock Log Sensitivity**

As mentioned in Section 2.4.2, "Delete Lock Log" is the capability least frequently permitted, and access should therefore be denied by default. Even for a spouse, this capability should not be accessed by default (only 68.4 % chose *always* for their spouse). More than 75 % of participants chose *never* for all other relationships. As the main method of retrospecting usage history, the log is not meant to be deleted.

B) **Supervising Children**

The elementary-school-age child (presented as 8 years old) was one of the most restricted relationships. On average across all capabilities, 69.4 % of participants chose *never* for the child ($\sigma = 0.19$, $median = 70.6\,\%$). Only neighbors received fewer permissions. In our chi-squared tests, we did not observe significant differences in desired access-control settings for children between participants who are currently living with a child, who have lived with a child before, and who have never lived with a child. None of our capabilities were considered child-friendly enough for even the majority of participants to *always* grant their elementary-school-age child access to that capability *always*. For only the "Light State" and "Play Music" capabilities was *never* chosen by fewer than half of participants. Despite being an immediate family member and living together, plenty of participants expressed fears that a child at that age might toy with these features and unintentionally mess up their settings or even cause danger to themselves. With supervision, though, many participants would consider giving temporary access to their children to gradually teach them how to use such a new technology.

C) **Ordering Online**

The capability to make an online purchase was generally limited to spouses only; 78.9 % of participants said that only their spouse should always be allowed to make online purchases, but 84.2 % also said that it was acceptable for non-spouse users to do the same if given explicit permission by the homeowner.

D) **Administrative Capabilities**

By default, only spouses should be able to access administrative capabilities, such as adding users, connecting new devices, and installing software updates. 89.7 % of participants gave their spouse access to these administrative capabilities *always*, while only 39.7 % of participants *always* gave comparable access to their teenage child. Unsurprisingly, under twenty percent of participants would give full access to other relationships.

2.4.5 The Impact of Context (RQ3)

Since there are many factors at play in the access-control-policy specification process, it is important to identify which contextual factors are most influential in this process and how they contribute to the final decision. The full results are visualized in Figure 2.3. We also ran chi-squared tests to see if each contextual factor had a relatively greater influence on some capabilities rather than others. While we did not observe significant differences for the "People Nearby", "Cost" and "Usage History" contextual factors across capabilities, we did

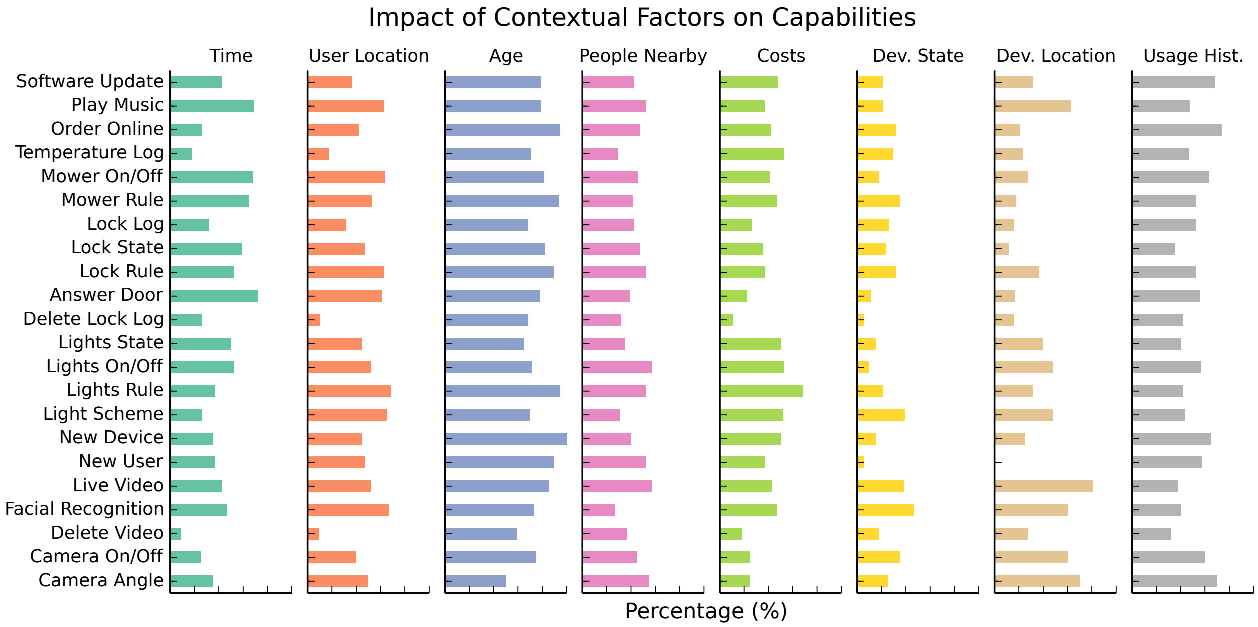


Figure 2.3: Contextual factors: Sometimes access must depend on the context. In the study we asked participants for such factors and identified multiple that are very influential (such as the age of the user) and learned how they contribute to the decision make process.

observe significant differences for the other five contextual factors.

A) Age

The *age* of the user was the most influential factor on average across the twenty-one capabilities, except changing camera's angle (78.1 % on average, $\sigma = 0.13$, $median = 78.3\,\%$). The proportion of participants for whom age mattered varied across capabilities ($p = 0.040$). The main capability for which age played less of a role was for *changing the camera angle* (only 50 %). Many participants were concerned with letting a young person have access to certain capabilities. *"They need to be mature enough to use it responsibly"* was one typical response. However, another participant instead explained, *"It will be the person themselves and how capable they are with technology. I do not care about age."*. Thus, while *age* was frequently mentioned, in reality the decision process is more likely to be driven by how capable and responsible a user is, which sometimes correlates with the user's age. Our results indicate that a child at a young age (around 8 years old) is generally not perceived to be tech-savvy and responsible enough to be allowed unsupervised access.

B) **Location of Device**

The proportion of participants for whom the device's location impacted the access-control policy varied across capabilities ($p < 0.001$). Capabilities relating to cameras were unsurprisingly very location-sensitive. "Camera Angle" is the only capability for which a device's *location* was more frequently influential (70 % of participants) than the user's *age. Device location* was the second most frequently invoked factor for turning a camera on or off (60 %) and watching live video (81 %). If a smart camera is installed indoors, especially in a bedroom or bathroom, it will be much more privacy-sensitive. Participants reflected this by saying, for example, *"I can see where a guest/house-sitter/baby-sitter might need to access a view of outside or the garage but not inside."* Therefore, when designing a smart camera, whether the camera will be used indoors or outdoors should be considered and reflected in default access-control policies.

C) **Recent Usage History**

The proportion of participants for whom a device's recent usage history impacted their access-control policy did not differ significantly across capabilities. On average across capabilities, 51.7 % of participants ($\sigma = 0.12$, $median = 52.6\,\%$) agreed that this factor impacted their decision about the access-control policy. For participants who felt the device's recent usage history would change their decision, two main rationales arose. On the one hand, if the history states that a user is abusing a capability, then the owner may revoke access. One participant wrote, *"If someone were to misuse the device, you best bet they aren't getting a second chance. Alright maybe I'll give them a second chance, but definitely not a third!"*. On the other hand, if a user turns out to be trustworthy, then the owner may consider letting them keep the access, or even extending it. *"If my kid had been using the device responsibly, I would feel more comfortable giving them more access."* However, some participants felt the recent usage history was not particularly relevant for two main reasons. First, if the involved capability itself cannot cause much trouble, such as “Light Scheme,” a common line of reasoning is that *“It would be hard to abuse this capability, so it doesn’t matter to me.”* Second, if the capability itself is so concerning that participants are reluctant to give others access (e.g., “Delete Video”), usage history did not play a role.

D) **Time of Day**

The importance of the *time of day* contextual factor varied across capabilities ($p = 0.001$). “Play music” (68.4 %) and lawnmower-related capabilities (64.7 % for creating rules for the mower, 68.2 % for turning lawn mower on/off remotely) were particularly sensitive to the time of the day. In order to not interrupt other people’s rest, participants tended to limit lawnmower usage usage to the daytime and playing music to the early evening.

E) **Location of User**

Capabilities that change devices’ behaviors tended to be more sensitive to where the user is physically located when trying to control the device ($p < 0.001$) since many functionalities cannot be enjoyed without proximity. For example, creating rules that control the lights (68.4 % of participants felt the user’s location mattered) and “Facial Recognition” (66.7 %) were prime examples. Many participants wrote that they would not want anyone who is not currently present in the house to use these capabilities unless it is the owner or their spouse.

F) **Costs**

The influence of the cost of exercising a capability did not vary across capabilities ($p = 0.162$). We believe this is in part due to our study design that did not include high-wattage appliances. Nevertheless, we observed some evidence of concerns with the cost of leaving lower-wattage devices, like lights, on during the day. Some participants mentioned that while lights do not consume a lot of electricity, cost can quickly become a concern if heavy appliances were to be involved. In addition, the influence of cost on online shopping differed due to different interpretations of cost. For cases where participants did indicate that cost is a concern, their interpretation was based on the cost of the good purchased, rather than the electricity used in placing an order.

G) **People Nearby**

43.6% of participants ($\sigma = 0.09$, $median = 43.6\,\%$) indicated that who else is nearby might impact their access-control decision. The role of people nearby did not differ significantly across capabilities ($p = 0.400$). For participants who believe this factor matters, there are two contrasting conclusions. Some people might feel more permissive when they themselves are around since that means they can supervise everything. However, others felt less permissive because if they are around, there is no need for others to have access since the others simply would need to ask the owner. Therefore, it is important for the system configuration to take these divergent mental models into consideration, letting users decide which direction they might choose to go in.

H) **State of Device**

The current state of device was overall the least important factor in participants' access-control decisions on average ($mean = 23.7\,\%$, $\sigma = 0.11$, $median = 22.3\,\%$), though this importance did differ across capabilities ($p = 0.044$). Notably, 46.7 % of participants who answered about the "Facial Recognition" the capability marked the state of the device as an influential factor. This is because if the camera is currently off, then there is no reason for anyone to enable of disable the facial recognition.

I) **Other Factors**

We included a free-text question with which participants could list other factors they thought played a role in their access-control-policy specification process. In their responses, we observed a long tail of additional contextual factors, including weather, people's familiarity with technology, how close they are to the owners, and the frequency of one's access to a certain capability.

2.4.6 Wrong Decisions' Consequences (RQ4)

Analyzing consequences of incorrect authorization decisions, we can learn how much tolerance a user has for a policy to fail given a specific capability and relationship pair. It is crucial to understand how strongly users would feel if the system were to malfunction. We analyze *false allow* and *false deny* decisions separately.

False Allow

Note that responses about *falsely allowing access* belong to those participants who intended never to grant access to a certain capability to a certain relationship. These participants therefore might be more concerned than other participants in certain aspects, which leads to some narrow tensions with the broader trends seen in previous sections. Figure 2.4 (top) summarizes these results.

A) **Neighbor false allows a major inconvenience**

Across all capabilities, 64.1 % of the participants stated that it is a *major inconvenience* if the authorization system gives access to their neighbor by accident. Turning the security camera on or off (100 % a major inconvenience) and creating rules for a smart lock (92.9 % a major inconvenience and 7.1 % a minor inconvenience) are the most concerning capabilities. Note that in the study, we described the people representing the relationship *neighbor* as "good people, which includes friendly small talk, and occasional dinner invitations." Nevertheless, privacy and security were major concerns.

B) **Spousal false allows have severe consequences**

Though the number of false-allow responses for the spouse relationship is quite small ($n = 10$), it still gives some interesting insights. 50 % of the answers are based on deleting log files from a smart lock. Four out of five respondents rate falsely allowing a spouse to delete the log file not to be an inconvenience. *"I wouldn't really care about my spouse deleting it, but it would bother me that the system is not secure,"* was a typical response.

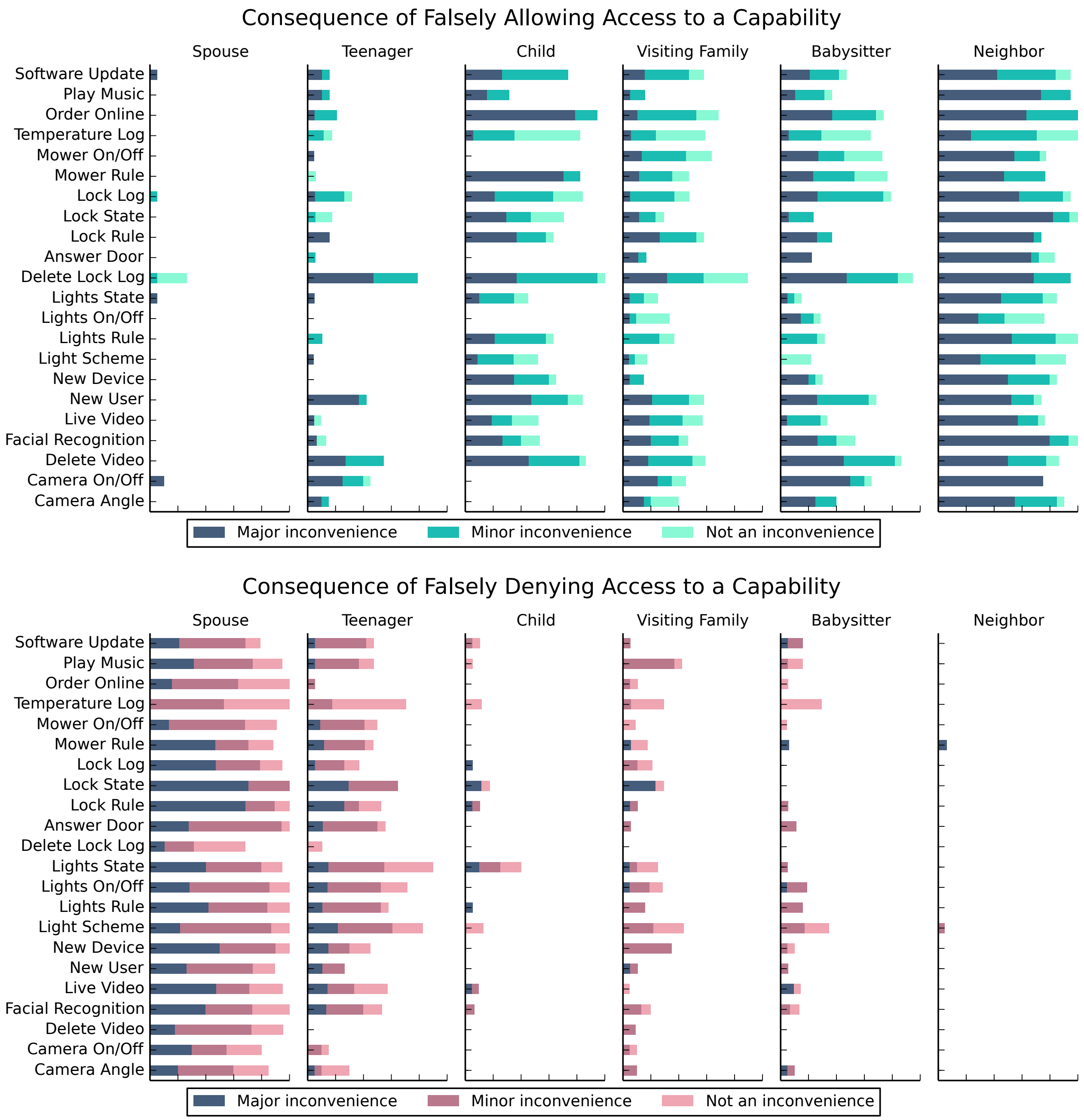


Figure 2.4: Perceived consequences of incorrectly allowing someone to use a capability when they should never be permitted to do so (top) or incorrectly denying someone when they should always be permitted to do so (bottom).

There were five more responses from other capabilities. From those, four out of five indicated that a false allow decision was a major inconvenience. It is surprising to see that a few participants believed it a major issue if the mechanism allows their spouse to access certain capabilities by mistake.

C) **Visiting family false allows a minor issue**

Though we presented earlier that participants' permissiveness toward a visiting family member and a babysitter was very similar (and tended toward not being permissive), we observed a distinction when it comes to false allows. Participants were much less concerned with incorrectly giving access to a visiting family member (70 % chose *minor* or *not an inconvenience*) than to a babysitter (58 %). Responses like *"He is my family member so I trust him a bit"* were common. While participants believed the visiting family member would not do much harm, false allows would still upset them a bit.

D) **Shopping / lawn mowers forbidden for children**

Among all capabilities, incorrectly allowing a young child to order online (79 % a *major inconvenience*) and create rules for the lawn mower (70.6 %) were the two capabilities where false allows for a child raised great concern. A child at such a young age is generally not trusted with ordering things online. *"The child could spend a ton of money on products we don't need,"* wrote one participant. A lawn mower is considered dangerous. One participant simply wrote, *"(A lawn mower) could cause harm to the child."*.

False Deny

Responses in this section, *falsely denying access*, come from participants who intended to give access to a certain relationship. Figure 2.4 (bottom) visualizes the full results.

A) **Participants Did Not Want to be Locked Out**

Lock-related capabilities raised the most concern (63.9 % of responses for "Lock State" and 58.8 % for "Lock Rule" found falsely denying access *major inconveniences*). Participants tended to be very cautious about smart locks. Even though viewing a lock state does not directly concern locking or unlocking the door, participants still worried whether a malfunctioning access-control system would lock people out, thus marking these false denies as major inconveniences.

B) **Spouses and Trust Issues**

One common reason why participants gave full access to their spouse is because they believe two people in a marriage should be equal, which means two parties should have the same access to a system. Therefore, if their spouse is accidentally rejected by the system, it could raise trust issues and spur arguments within the marriage. We found a number of responses similar to *"I would not want my spouse to think I did not trust them."* It is interesting to see that not only do relationships impact access-control policies, but relationships are also influenced by authorization results. Thus, extra care is required for such relationships.

2.5 Conclusion

Capabilities, Relationships, and Context. While access control in smart homes is currently often device-centric, our user study demonstrated that a capability- and relationship-centric model more closely fits user expectations. Home IoT technologies allow for multiple ways of achieving the same end result, whereas devices often bring together vastly different capabilities. For example, to increase a room's brightness, one could remotely turn on a light using a smartphone app, remotely open the shades, or ask a voice assistant to do either. This model reveals nuances that are missed in the device-centric model. From the data for RQ1, we see that the desired policies can vary widely within a *single* device based on the relationship and the context of access. Although some of these distinctions are intuitive (e. g., child vs. teenager), others are more nuanced and surprising (e. g., babysitter vs. visiting family member). They also provide a concrete access-control vocabulary for developers of future smart-home devices.

A difficult decision in access-control systems involves default policies. In multi-user social environments, intuition suggests a default policy would be complex. Surprisingly, our data for RQ2 suggests that potential default policies are actually simple and reminiscent of non-

IoT policies. For example, our default policy says that a person can actuate a light if they are physically close to it. Though IoT lights can be remotely actuated, the relation between proximity and using a light is not broken. Although conceptually simple, this rule's enforcement is non-trivial, requiring creating and deploying authentication methods beyond the possession of a smartphone.

Data from RQ3 suggests that the factors affecting access-control decisions are heavily context-dependent. Current home IoT devices only support rudimentary forms of context (Section 2.1). Some contextual factors, such as age, are currently present in smartphones and cloud services (e. g., *Apple's iCloud Family Sharing* supports adding a child Apple ID that requires parental approval for purchases, while Netflix has *kids* option). We recommend that for home IoT settings, these contextual factors should be a first-order primitive.

Based on these findings (RQ1-3), we envision several changes to smart-home setup. This process currently involves installing hubs and devices with a set of coarse-grained accounts. Our work suggests that future smart homes could instead set access-control policies by walking users through a questionnaire whose vocabulary derives from our user study. This is closer to the experience of setting up software, where a package comes with secure defaults that are customized to the specific installation. Using default policies derived from our results would minimize user burden since it would reflect common opinions by default. Physical control (e.g., switches) already enables certain default policies, so software authorization might seem unnecessary in certain situations. However, switches are often add-ons to IoT starter kits, making software authorization a prerequisite to a satisfying user experience.

Authorization Vocabulary. Based on our study results, we discuss a potential authorization vocabulary that is helpful in building future authorization and authentication for home IoT platforms. The basic unit of the vocabulary is a triplet containing <Capability, UserType, Context>. As discussed, capabilities better capture the nuances of access control in the home than devices. UserType captures the relationship of the user to the home, and to the owners. From our study, these types should nonexhaustively include: Spouse, Teenager, Child, Babysitter, and Neighbor. Spouses tend to be users with the highest levels of access, generally equivalent to administrators in traditional computing systems. Context refers to the environmental factors that might affect an access-control decision. For example, certain parents might be more permissive in allowing a child to watch TV without supervision. Based on our study, at the minimum context should include: Time, User Location, Age, People Nearby, Cost of Resource, Device State, Device Location, and Usage History. Depending on the Capability and the UserType components of the triplet, the importance of the context can change. For example, for a UserType of Child, the 'People Nearby' contextual factor plays a prominent role in the access-control decision. However, for spouses, it generally has no bearing. The same goes for the Capability. The 'Device Location' contextual factor is crucial for camera-related capabilities, but not so important for the capability of adding a new user.

Mapping Authorization and Authentication. Although we focused on analyzing access control, we briefly discuss how our findings affect the design of authentication mechanisms. Below, we discuss a set of authentication mechanisms and comment on their ability to identify users, relationships, and contextual factors. We also discuss privacy limitations and the effect of false positive and negatives.

Smartphones are the most widely used devices to access IoT devices in the home. Users may present their identity to a device using a password, PIN, or (more recently) fingerprints. These identities can be used by home IoT devices to determine the identity, and hence relationship, of the person attempting access. From the perspective of false positives/negatives, smartphones can closely match user expectations. They are inconvenient, however, for temporary visitors because they require the visitor to install an app and the owner to authorize them.

Wearable devices like watches, glasses, and even clothing [43] might serve as proxy devices with more natural interactions than a smartphone. For example, a user can gesture at a nearby device to control it (e. g., wave at a light to turn it on or off). As each user will perform a gesture differently, it can also serve as a form of authentication and thus be used to identify a person and their relationship. Furthermore, the proximity of a wearable device is helpful in identifying several contextual factors, including user location and nearby people. From a false positive/negative perspective, biometrics require quite a bit of tuning that can affect an owner's choice of using this method, especially when authenticating high-access spouses or for operating dangerous equipment like lawn mowers.

Voice assistants are increasingly ubiquitous in homes. Although such assistants can perform speaker identification (e.g., Google Home Voice Match), they are currently used as a personalization hint rather than a security boundary. However, future versions that use additional hardware might be useful in determining a speaker's identity and relationship for access-control purposes [44]. Such assistants could help identify contextual factors like the location of a user or the presence of nearby people (e.g., a supervising adult near children). From the perspective of false positives/negatives, any voice-based method will require tuning. Audio is especially sensitive to background noise. Audio authentication also introduces privacy issues, as well as the potential for eavesdropping and replay attacks. Advances in computer vision can also be leveraged to identify users, their relationship, and their location within a home with cameras. However, it is possible for computer vision systems to falsely identify individuals or confuse identities. Thus, some level of false positive/negative tuning will be required, especially when a household is expected to have many temporary occupants. A big downside of this mechanism is the privacy risk—cameras can track home activity at a high level of granularity. However, some of the privacy issues could potentially be alleviated using local processing or privacy-preserving vision algorithms [45].

Bilateral or continuous authentication mechanisms embody the idea that a user has to be: (a) physically present, and (b) currently using the device [46, 47]. Such mechanisms are readily able to identify users and relationships, and to support contextual factors involving user presence. False positive/negative tuning varies based on the specific instantiation. If a wearable device with a continuous authentication algorithm is used, then the false positive/negative rates must be considered. Privacy concerns can be alleviated if this mechanism is implemented in a decentralized manner—only the user's proxy device and the target device are involved in establishing an authenticated channel. It can also provide a simple solution to the de-authentication problem (revoking access if a temporary visitor is no longer welcome).

In sum, we have taken initial steps toward reenvisioning access-control specification and authentication in the home IoT. Much work remains in continuing to translate these observations to fully usable prototypes, as well as in supporting ever richer capabilities and interactions.

CHAPTER 3

CONTEXT SENSING FOR ACCESS CONTROL IN THE ADVERSARIAL HOME IOT

As discussed in the previous chapter, in home IoT, the set of users who ought to have access to a resource varies over time and may include guests, in-home workers, and others [2, 48, 49]. Desired access control in smart homes is frequently *contextual* (situational). Rather than granting unconditional access to a given user or a given role, authorization decisions may depend on the context. A context can be the user's location relative to the device, the history of the user's interactions with the device, or the state of the home [8]. An example policy is that a child can only use the smart TV when a parent is nearby [8]. Here, the system must verify two contexts: (i) a child is trying to use the TV and (ii) a parent is around. Enforcing contextual access control requires privacy-preserving and trustworthy context sensing. That is, a *sensor* (e.g., a motion sensor) must reliably detect some *context* (e.g., a room is unoccupied) while respecting users' privacy.

Prior work in the security and privacy community has already proposed ways to utilize contexts in access control [50, 51], but has not focused on how to detect contexts in the physical world in ways that are both trustworthy and privacy-preserving. A large amount of existing work on sensing and ubiquitous computing could be applied here, but it mostly ignores attacks, adversaries, and privacy. For example, work done on robust sensing often sacrifices privacy by adopting more invasive sensing methods [52] or denser sensor deployment [53, 54]. This is not realistic for an intimate setting like one's home. Some bodies of work also discover that errors are bound to occur in particular circumstances, but they regard these errors as rare or unintentional occurrences [55, 56, 57]. Adversaries can exploit this assumption.

In this chapter, we critically reevaluate the literature on context sensing in homes with a security and privacy mindset. Furthermore, we translate this literature to the problem of context sensing for access control, identifying sensor types that best match specific contexts within practical constraints. To do so, we first identified home contexts that are critical to access control from the small literature on contextual access control in smart homes. We then systematically searched the proceedings of the last decade of top conferences in sensing systems (SenSys, MobiSys, and MobiCom), ubiquitous computing (UbiComp/IMWUT), and human-computer interaction (CHI and UIST), identifying dozens of recent papers about sensors that can detect those contexts in smart homes. To capture well-known mature sensors, we also searched for commercially available sensors for smart homes and added classic papers on relevant sensors. This process left us with 94 pairs of contexts and sensors. Analyzing these papers while also revisiting key IoT papers from the security, HCI, and usable security literatures, we constructed a decision framework that highlights each sensor's pros and cons for security, privacy, and usability when used to detect an access-control-relevant context in a smart home. Our work thus lays a foundation for secure, practical, and privacy-preserving context sensing in smart homes.

We first create a novel threat model broadening the adversaries that prior literature has considered for smart home sensing. Prior work has focused on how experts can exploit IoT systems through software vulnerabilities [7, 5], default passwords [32], replication of physical traits [58], and adversarial examples [59, 18, 20, 19]. While our model encompasses these threats, we focus on non-technical adversaries with legitimate access to a home, such as kids, roommates, guests, and workers, who usually have stronger motivations than remote strangers. Notably, most papers on context awareness and home sensing do not consider the adversarial mindset typical in the security community.

From our threat model, we make several observations. First, physical denial-of-service attacks are trivial against many sensors. Thus, in contextual access control, policies that allow access by default or rely on the absence (rather than presence) of a characteristic are easy to bypass. Second, non-technical users are highly capable of replay, imitation, and shoulder-surfing attacks. They can also impersonate someone by simply taking that person's phone. Identity cannot be reliably authenticated through possession of a phone or naive recognition of voices/faces.

Contextual access control in homes thus requires deploying sensors with key properties. The sensors, alone or in ensemble [53], must resist attacks from both technically literate outsiders and non-technical insiders. They must also minimize inadvertent data collection because sensors may be deployed in private areas of the home. Finally, household members must find the sensors acceptable.

Then, we develop a decision framework for evaluating the degree to which a particular sensor possesses these key security, privacy, and usability properties. We further distinguish between attacks of different complexities, privacy considerations from various actors, and specific usability criteria. The latter includes ease of deployment, reusability of a sensor across contexts, and inclusiveness. This framework will be useful for individuals who design or deploy sensors in homes, including DIY users [60], manufacturers, and researchers in security and in sensing. We will refer to these individuals as *smart home designers*. This framework can help smart home designers navigate the vast array of sensing mechanisms described in the literature or available commercially. We envision the framework helping smart home owners to decide which sensor to use, manufacturers to design their products for facilitating contextual access control, and researchers to develop sensors that are more sensitive to security and privacy issues. The framework also outlines criteria to consider when designing a new sensor. In particular, our framework elucidates key trade-offs among the variety of sensors (e.g., motion sensors, microphones, thermal imaging) that can detect a given context (e.g., whether anyone is in a room).

Eventually, we apply our framework to highlight trade-offs in deploying sensors for access control in homes. Through a systematic review of the sensing literature, we identify *indicators* (e.g., characteristics, such as gait) and associated *sensors* (e.g., a pressure sensor mat for detecting gait) for sensing either *identity* (e.g., this is Jane) or *context* (e.g., this is an adult). Using our decision framework, we evaluate each sensor's key properties. We used our literature review to gauge sensors' robustness to attack, privacy properties (e.g., requirements for data storage), and deployability. With our framework, smart home designers can identify the sensors that support desired contexts for access control and recognize trade-offs in security, privacy, and usability. **To keep our framework and evaluations up-to-date, we have released them in a public GitHub repository.[1]** Researchers may publicly modify, expand, or dispute the table through pull requests and issues, facilitating open discussion between the sensing and security communities.

Applying our framework yields the following insights. First, we find that many current sensors, when used alone, do not adequately address potential threats from non-technical adversaries. They are especially vulnerable against rarely studied physical DoS attacks. Second, many sensors collect more data than needed. Contrary to currently deployed architectures, many sensors do not require cloud storage for data. Lastly, we found that many sensors are not inclusive based on age or disability, and some can be ineffective under certain environmental factors.

3.1 Smart Home Model

Context sensing and access control depend heavily on how a smart home works. Here, we abstract away implementation differences and discuss a model that applies to most smart homes. Current IoT devices support rich functionality, yet access control in the home has largely been limited to using smartphones as a proxy for identity.

Figure 3.1 depicts our basic model. A smart home consists of two types of Internet-connected devices: *actuators* that execute commands (e.g., lights), and *sensors* that measure their surroundings (e.g., motion sensors). Users control actuators through *interaction*

[1] `https://github.com/UChicagoSUPERgroup/eurosp21`

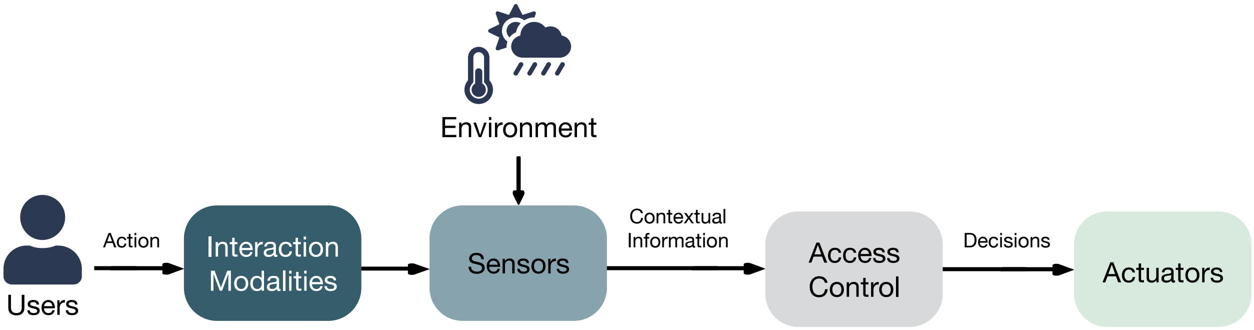


Figure 3.1: Our model of a smart home.

modalities (e.g., smartphone, voice, physical buttons). The access-control policy uses contexts sensed via sensors to decide whether to authorize access.

Actuators can be controlled over the internet or a local network, enabling access control [50]. Traditional devices (e.g., non-IoT locks) are outside our model.

Users are people with remote or local access to devices, including family members, visitors, and workers.

Interaction Modalities describe how the user interacts with devices. Our model includes five modalities. The first four typically result in immediate changes, while the last covers automation that causes future changes.

1. Manual Interaction: A user can interact with devices manually, often by flipping switches or pressing buttons. Additional sensing is required to identify the user in such scenarios. A contextual access-control framework can inform a smart device whether to permit access.

2. Smartphones: Smartphone apps can control devices, sometimes via a home hub. Because users already authenticate to their phone, current IoT systems often rely on the possession of a phone as a proxy for identity.

3. Voice: Voice assistants let users control devices by speaking. Currently, they perform no authentication [17] or use speaker recognition that is easy to fool [61, 62].

4. Gestures: Currently uncommon in homes, gestures could be detected using ultrasonic or radio waves to recognize and authenticate movements as a source of input.

5. Automation: Smart home automation can link changes in context or other triggers to actions. They can be set with apps [63] or end-user programming [64]. Absent access control, automations may create loopholes [65, 23]. Imagine the automation: “If the lights turn off then play a movie.” If a child may not play movies, yet may turn off lights, a crafty child could start a movie by turning off a light. While focused on contextual access control, our framework can also apply to automations triggered by a sensed context [64], such as when a room is warm [66, 23]. An attacker who tricks a sensor can cause chained automations toward a malicious goal [65, 35].

Contexts describe a particular state of the physical world. In a smart home, contexts describe situations, states of actuators, presence of specific people, and more. Examples include a security camera being activated, the temperature staying within some range, or a specific person sitting in the kitchen. Contextual access control relies on sensors to reconstruct these situations.

Sensors detect physical properties. Traditionally, they have been used primarily for smart home automation (e.g., motion triggers a light). However, recent research has identified the need for contextual access control in the smart home [50, 8, 11]. We envision that both existing and future sensors will underpin this paradigm.

Smart homes use phones or accounts as an imperfect proxy for identity. Context sensing has generally been used for automation, not contextual access control.

3.2 Our Threat Model

Sensor-based access control in homes requires robust sensing that protects user privacy. Prior IoT research has primarily focused on defending against remote attacks against IoT software [7, 6]. However, local attackers—regardless of technical background—can also pose a significant threat to the system by tricking physical sensors into detecting incorrect contexts or violating others’ privacy. In fact, potential local attackers like family members, roommates, guests, and workers could have stronger motivations to bypass access control than unacquainted remote attackers. Our work examines local threats broadly and focuses on those posed by non-technical users with legitimate or illegitimate access to a home. Below, we taxonomize goals, attacks, and attackers. In light of the larger literature on context sensing, we revisit these attacks within our decision framework (Section 3.3).

3.2.1 The Attacker's Goals

One of our key insights is that non-technical attackers with modest and localized goals are a threat to contextual access control. Whereas remote attackers disrupt at scale, non-technical local attackers might only want to gain illegitimate access to some resource or spy on another individual. For example, a child may wish to watch TV without approval, a burglar may want to erase security camera footage after committing theft, or (as can be the case with intimate partner violence [67, 37]) an abusive member of the household may try to spy on members of their household by evading policies stopping security cameras from recording when people are home.

Local attackers might aim to bypass access control or compromise the privacy of others in the home.

Strategies for attacking sensors depend on the policy. A *default-deny* policy, which automatically denies access to unknown users, is not always advisable. For instance, prior work found users prefer default-deny policies for locks, but would rather permit unauthorized users to control smart lights than leave users in the dark [8].

Impersonation: Under a default-deny policy, a system only accepts authorized and authenticated users. An attacker must impersonate an authorized user or fabricate a valid token through imitation or replay attacks.

We find that these attacks often do not require technical knowledge (Section 3.5), especially in an intimate setting like a home where boundaries to privacy are reduced and private resources are easy to acquire. For example, many widely deployed facial-recognition systems lack depth or liveness detection. One can trick them by presenting a photo or video of an authorized user [68]. Photos of authorized users (e.g., a child's parents) are easy to find in a home, and videos can be taken in secret.

Similar issues arise for audio. People with access to a home can record authorized individuals speaking to voice interfaces. While authenticated speaker recognition is an active area of research [69], many widely deployed voice interfaces are vulnerable to simple replay attacks [61, 62] or even lack authentication entirely [17].[2] Off-the-shelf voice morphing compounds this problem [70].

Local attackers have extensive access to photos and audio, making basic face or speaker recognition systems vulnerable to replay and imitation attacks.

Current home IoT systems tend to rely on smartphones as a proxy for identity, capitalizing on their ubiquity. However, smartphones often run out of battery, and they do not offer the convenience of other interaction modalities (Section 3.1). This practice also falsely assumes that the user is always near their phone. For example, if the smart TV will turn on only if an adult's phone is in the room, a mischievous child can take their parent's phone while the parent is sleeping. Furthermore, smartphone authentication is still not fool-proof as it is often knowledge-based (e.g., PINs). It is often easy for others in the home to bypass this authentication through shoulder-surfing.

Existing practices of using phones (potentially with authentication) as a proxy for identity in shared spaces can be risky in terms of both security and usability.

Invisibility: Contextual access-control policies can also allow access by default. One example would be using the smart stove. Whereas visitors or babysitters may be allowed to use the stove, a child should not use it for safety reasons. A natural policy that follows is

[2] In our informal testing, Google Home's speaker recognition only seemed to verify the person who said "OK, Google." It accepted further commands spoken by someone else, making replay attacks trivial.

Dimension	**Type**	**Capabilities**	**Examples**
Access	Indoors	Physical access to indoor & outdoor devices/sensors Rich observation opportunities Full knowledge of sensor models & locations Knowledge of access-control policies & automations	Family member, babysitter
	Outdoors	Physical access only to outdoor devices/sensors Limited observation opportunities Opportunistic attacks that reach more victims	Neighbor, prospective burglar
Expertise	Expert	Sophisticated network and imitation attacks Ability to craft black-box adversarial examples Unsophisticated replay/imitation attacks, block sensor	IT professional, hacker
	Non-expert	Unsophisticated replay/imitation attacks, block sensor	Child, domestic worker
Resemblance	Similar	Spoofing (through imitation) Higher possibility of inadvertent false positives	Sibling, one who looks similar

Table 3.1: Local attackers can be characterized along the dimensions above, impacting attack capabilities.

“anyone except a child can turn on the stove.” When these *default-allow* policies depend on *not* sensing a characteristic or situation, e.g. “record security video of the bedroom when *no one is home*), an attacker needs nothing more than to make the characteristic or situation “invisible.” They can do this by changing or blocking the sensor’s field of view.

We will refer to such attacks, where the local attacker prevents the sensor from physically detecting a context, as *physical denial of service (DoS)*. This can entail blocking a motion sensor with paper or overloading a microphone with loud noise (including outside the human hearing range [71, 19]). Sensors must detect whether they are receiving accurate and fresh input.

Default-allow policies, which rely on *not* detecting a given situation, can be defeated by blocking sensors.

3.2.2 Attacks

Based on these attacker goals, we surveyed top security and sensing conferences to identify likely attacks. We clustered prior work based on attack method, resulting in three major types of attacks: 1) replay and spoofing attacks; 2) adversarial examples; 3) sensor hardware attacks. Note that replay and spoofing attacks differ in practicality despite often appearing together in the literature. We did not find mentions of physical DoS attacks in our literature survey, but include them because they are a clear threat to access control. Below, we define these attacks.

Replay Attack: The attacker collects a credential and feeds it back to a sensor. For example, the attacker can play a voice recording, show a photo of a face, or make a gummy mold of a specific fingerprint [58]. Our focus in this SoK is on replaying the physical signal itself, although network traffic can sometimes also be replayed.

Spoofing: The attacker forges an approximate credential or situation they have not necessarily captured. Smoke can spoof a fire, and energetic pet cats can spoof occupancy.

Physical Denial of Service (DoS): Jamming, blocking, or moving a sensor can prevent accurate sensing. It is important to note that the sensor detecting the *absence* of a characteristic or situation is different from *not* detecting it. For instance, when trying to sense whether a room is empty, a camera blocked by a piece of paper will not detect any people. This differs from a camera affirmatively seeing a room without people. These attacks are often easy to deploy, but have not yet received much attention.

Adversarial Examples: Against ML-based sensing methods, the attacker can poison the training data or add carefully crafted noise to inputs [59, 72].

Sensor Hardware Attacks: The attacker leverages the physical principle behind the hardware to deceive the sensor, such as with signal injection attacks [19, 73].

Inadvertent False Positives: This is not quite an attack, but a sensor incorrectly detecting an identity or situation can still compromise access control.

3.2.3 Physical Sensors' Potential Attackers

To understand each attack's feasibility, we characterize the attacker's capabilities. Table 3.1 provides a summary. Our threat model concerns attackers who *violate* access-control policies. We thus ignore adversaries who *create* unreasonable policies, such as domestic abusers attempting to spy on their family. Defending against those adversaries requires countermeasures beyond access control.

Access: An attacker with access to the home would be well-positioned for physical attacks against sensors. They can observe authentication processes in the home, potentially repeatedly, to record information for replay or imitation attacks. For example, a roommate might encounter multiple instances of the user speaking to a voice assistant. They thus have multiple opportunities to record the user's voice for tricking speaker-recognition algorithms. By having access to the home, attackers can also infer access-control policies, automations, and sensor locations or types from their observations. Legitimate access can be permanent, such as for residents, or temporary, such as for visitors and domestic workers. Illegitimate access occurs when people enter the home without permission.

It is also possible for attackers to access sensors outside the home [74, 13] or make inferences using partial information (e.g., from sensors visible through windows). Some individuals who might rely on these methods include neighbors and prospective burglars. We note that modeling the attack surface cannot rely on a simple indoor versus outdoor dichotomy. For example, one can control a voice assistant through an open window.

Expertise: Attackers with technical expertise, such as infosec professionals, are capable of sophisticated attacks. Some attacks against ML-based sensor systems are of this nature. They can involve carefully crafted eyeglasses [59], stickers [18], or audio [71, 19]. Experts can also target sensors' physical principles, such as applying acoustic interference to accelerometers [75]. Finally, network- and software-based attacks are also possible.

On the other hand, nontechnical attackers can carry out replay or imitation attacks that only require observations (e.g., spoken passwords) or commodity recording equipment (e.g., a smartphone). They can also disable sensors by blocking, repositioning, or unplugging them.

Resemblance: Biometric sensors may confuse individuals of similar physical traits. Biological family members often share physical resemblances and have easy access to sensors because they often live together or visit each other. Real-world examples include one man who tricked a voice-recognition system by imitating his twin's voice [61]. Identical twins can also fool facial recognition [76]. It may also be possible for unrelated people with physical resemblances to trick the sensors.

Our threat model highlights two key ideas missing from prior work. First, most work focuses on threats from attackers with extensive resources and expertise. We show that non-experts with access to the home are capable of replay and spoofing attacks against sensors that support contextual access control. Second, blocking sensors can allow attackers to evade some access-control policies. This method of attack has not yet been studied extensively.

> Contextual access control must consider that non-experts with access to a home can attack sensors.

3.3 Decision Framework for Context Sensing

Individuals designing or deploying home sensors need a framework that helps them navigate the trade-offs between sensors' security, privacy, and usability properties in conjunction with the users' needs and the space itself [77]. These individuals, whom we term *smart home designers*, will benefit from the framework in different ways:

- Do-it-yourself smart home owners can learn security and privacy implications of selecting certain sensors.
- Sensor manufacturers can holistically evaluate their current sensors' trade-offs and identify additional contexts that need new sensors to be developed.
- Security and sensing researchers can identify security and privacy gaps that guide their future research.

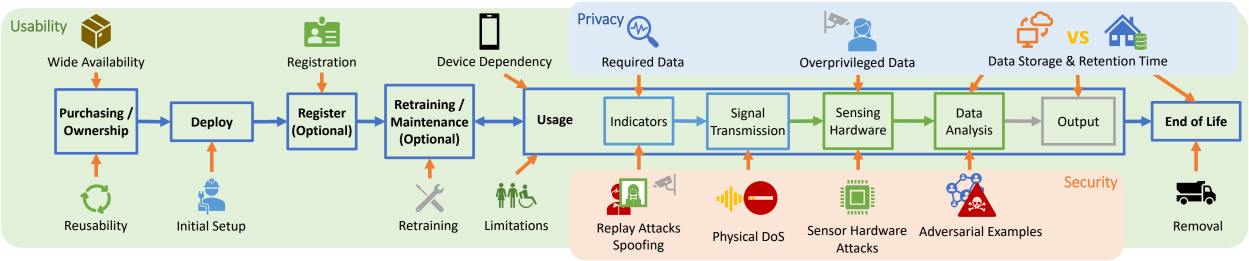


Figure 3.2: Different issues emerge in difference stages of using sensors in home.

For example, a smart home owner might wish to know when anyone is at home. Consulting our framework reveals that cameras are suitable for this, but are not privacy-preserving. Meanwhile, pressure sensors on the floor would be privacy-preserving, but are impractical and expensive to install. The user can now determine whether to prioritize occupancy detection at the cost of privacy.

Here, we first explore the life cycles of adopting a sensing technique. Then, for each stage of the life cycle, we further define the main security, privacy, and usability criteria that smart home designers must consider in choosing sensors, which we collectively consider our framework. We constructed this framework by critically analyzing the 94 pairs of sensors and contexts we identified through our systematic review of the sensing literature (see Section 3.4.2) relative to the security and usable security literatures concerning the home IoT. We also considered broader security principles to fill in potential gaps in this framework.[3]

3.3.1 Life cycles

Adopting a new sensing technology in one's home is a long-term and ongoing process. To avoid missing crucial challenges during the process, we first define different stages of the adoption process, as depicted in Figure 3.2.

Acquiring the required hardware: A user might need to buy new sensors, which is a

[3] The team that constructed the framework included multiple students and three faculty members. Two of the faculty members focus on security and privacy research, but also have experience with machine learning research. The other faculty member conducts sensing research.

financial and time investment.

Deploying the hardware: After acquiring the hardware, users need to install it in their homes. When needed, users might also re-deploy hardware, such as to reposition it.

Registration (optional): Sometimes the hardware may require the user to register themselves first, which is especially common for sensors pertaining to an identity.

(Re)training / Maintenance (optional): Before usage, machine learning-based sensing methods commonly require the user to train the model about the context in its unique environment. Retraining may also be required in the future to adapt to users and a sensor's environment changing over time. Maintenance, such as battery replacement and routine check-ups, may also be required.

Usage: After training, the sensor is ready for use. We expect the sensing technique to operate until the user stops using the sensor. To identify possible issues in this stage, we must abstract how the sensing technique works.

Sensing detects environmental events, such as temperature changes, movement in the background, and sound. We term these *indicators*, which could be mapped to a context. For example, if a sensor detects movement of a heat source, it is likely to be someone moving nearby.

To detect the indicator, the sensor needs a signal sent or radiated from the source. Depending on how far the signal can be transmitted, the sensor may require direct contact, near-field communication, or far-field communication. We term this process *signal transmission.*

Once the sensing hardware receives the signal, it first needs to process the analog signal, such as using amplification and noise filtering. The analog signal can then be converted into a digital signal for further processing. The *sensing hardware* stage represents the above process.

Finally, the digital signal, or the raw sensor data, is sent to a processor or the cloud for further computation. Depending on what the sensing method is designed for, different *data analysis* methods may apply here. For example, facial recognition and gait recognition may both rely on cameras, but the data analysis would differ. Once the sensed data analysis is complete, the algorithm *outputs* whether the context it aims to detect is active.

End of Life: The user may eventually decide to uninstall the home sensor. In this stage, the sensor may be directly thrown away, given to others, or sent back to the manufacturer for upgrading or replacing. The hardware is not guaranteed to be properly destroyed. Thus, information leakage after disposal is possible. We treat the uninstallation process as two parts: removing all data (e.g., factory reset) and physically removing the sensor from the home.

3.3.2 Security

We consider two ways in which a sensor may be attacked. One is through *inadvertent failures*. An attacker may bypass an error-prone sensor through brute force. The other is through *intentional attacks*. These attacks are described in detail in Section 3.2. Figure 3.2 also indicates at which stage these attacks might occur.

We do not consider attacks before the usage stage. The set-up stage occurs only once and the victim is often present, increasing the difficulty of attacking the sensor itself. Therefore, during the set-up stage, it is more likely for the attacker to perform network attacks (e.g., sniffing, person-in-the-middle), which are out of this paper's scope.

In Tables 3.2-3.4, a red "!" signifies that a sensor is easily susceptible to a given attack. A yellow "?" signifies that it is not very susceptible to the attack. If no symbol is shown in the table, the attack is implausible against the sensor (e.g., replay attacks against smoke detectors).

3.3.3 Privacy

Sensors collect data to operate, but excessive collection of sensitive data causes privacy concerns. Furthermore, certain contexts require intensive computation on data that is collected over long periods of time. To identify potential privacy threats during the usage stage, we review each stage carefully to identify general threats. We assume that the sensing software is secure and do not consider privacy threats before the usage stage. Our framework considers the following aspects:

Required Data: Data that must be collected for the sensor to function. Depending on which *indicator* the sensor detects, different types of data are collected, with various privacy implications.

Overprivileged Data: Depending on which sensor the designer decides to use, superfluous data might be collected inadvertently. For example, a microphone for occupancy detection also records conversations. In the “overprivileged data” column of Tables 3.2-3.4, *poor* means the sensor collects unnecessary and sensitive information, *acceptable* means it collects unnecessary data that is not sensitive, and *good* means it does not collect superfluous data.

Data Storage: Data must be analyzed and stored in the cloud if the device lacks the computational power or storage space for local processing. For other sensors, however, data can be stored on the device containing the sensor or on an in-home hub. Nonetheless, companies often upload data to the cloud even when unnecessary [78]. There is no guarantee that the uploaded data will be used ethically [79], which can deter users from deploying some sensors in homes [77]. We consider whether each sensor’s data *must* be stored on the *cloud*, or whether *local* storage supports the needed functionality. We leave out of scope the question of whether a company will choose to upload data to the cloud even when it could be retained locally.

Retention Time: Some sensors require longitudinal data (e.g., for training a model). Companies may again decide to store all data indefinitely even when not strictly necessary. *Tran-*

sient storage means sensed data can be immediately discarded, while *persistent* means it must be retained until the user factory resets the device. Similar to *data storage*, companies may retain users' data for as long as they want, even if the user factory resets their device and deletes their account. To focus on the requirement for enabling the sensing technology, we only consider how long the data must be available for the functionality.

3.3.4 Usability

To assess a sensor's usability for a non-technical end user, we consider the following criteria, which we compiled based on the stages identified in Figure 3.2.

Wide Availability: Users are more likely to adopt sensors that they can easily acquire. For example, one can sense occupancy with motion sensors or ultrasonic sensors, but users and designers may prefer the former because of their cheap cost and ubiquity. Nonetheless, more expensive sensors (e.g., cameras) may also be widely available if they fulfill multiple use cases. This may benefit users because sensors that fulfill multiple use cases may obviate the purchase of additional sensors.

Initial Set-up: How difficult is it for a non-technical user to set up the hardware during the deployment stage? *Good* means little to no effort is required, such as plug-and-play installation. *Poor* requires substantial effort from the user, such as renovating their current home for installation (e.g., painting the wall, changing the floor). Anything between *good* and *poor* was deemed *acceptable*.

Registration: How much effort does it take to register a user, or how long does it take to collect enough data to train the model? *Good* means no registration or training is needed. *Acceptable* encompasses two situations. In the first situation, the sensing method requires straightforward registration or data collection, meaning registration should not take over 10 minutes. This includes most commercial products, such as Touch ID or Face ID. In the second situation, data collection needs more time to finish, but does not require user attention. For example, a system from Hsu et al. [56] required the user to wear an accelerometer for days as ground truth for identifying the user from their RF reflection. While this process takes days, no attention is required, earning it an *acceptable* rating. *Poor* takes significant effort from users, usually exceeding 10 minutes in duration while requiring constant attention the entire time. For example, Qian et al.'s system [80] requires the user to walk for four minutes each at three different paces.

Retraining / Maintenance: How often is model retraining or hardware maintenance required? *Good* requires none. *Acceptable* requires occasional retraining or maintenance less than once a month (e.g., changing batteries every few months). *Poor* requires retraining or maintenance at least once a month. When evaluating biometric sensors, we assume an adult user with stable features.

Reusability: Some sensors can detect multiple contexts. For example, cameras can detect age, room occupancy, or an identity. *Good* means many contexts can be sensed, as with cameras. *Acceptable* means a few contexts can be sensed, as with radar sensors. *Poor* means the sensor detects only one context, as with fingerprint sensors.

Device Dependency: Some methods require users to carry a device (e.g., a phone) during usage. *Good* means no such device is required. *Poor* means that it is required.

Limitations: We consider whether the sensor is effective for all groups of users and under all situations. We focus on age, potential disabilities, and environmental factors (e.g., lighting conditions, GPS reception underground).

Removal: When a user decides to stop using a sensor, the sensor will be removed from the home. As removal is the inverse of the initial setup, we decide to combine them with the initial setup in Tables 3.2-3.4.

3.3.5 Example

We illustrate the use of this framework by describing two examples. Both examples are sensors that one might use to detect robbery, which is relevant to when access is granted based on whether there is an emergency in the home. They are also listed in Table 3.2.

Some commercial products, such as the Netatmo Camera [81], alert the user when unrecognized individuals enter the house. As one would expect, cameras and facial recognition algorithms have poor security and privacy qualities, but great usability. They are easily susceptible to replay attacks and adversarial examples. They are also susceptible to physical DoS if the attacker simply blocks the field of vision with an object. Sensor hardware attacks and spoofing are likely impossible for the adversaries we consider. The video stream will capture more information than needed to determine the occurrence of a robbery. Processing the video stream requires long-term cloud storage. Lastly, cameras are ubiquitous and easy to use, although registering users and retraining the facial recognition algorithm to accurately recognize users require some effort.

Glassbreak sensors, like Honeywell's [82], can also detect robbery by monitoring for audio frequencies of glass breaking. These sensors are susceptible to replay attacks, physical DoS, sensor hardware attacks, and spoofing. Machine learning is not necessary, so adversarial examples are not a concern. They capture basic audio frequencies that encode more information than necessary, but this information is simple enough to be stored locally for a short amount of time. They are easy to acquire and use, but they only fulfill the unique purpose of detecting glass breaking. A user looking to sense multiple contexts cannot rely on glassbreak sensors for other contexts.

3.4 Methodology

Both to understand the potential of applying our decision framework in realistic situations and to illustrate how to use it, we applied the framework to sensors that would support commonly desired contextual access control policies in smart homes. Applying the framework requires: (1) a set of desirable contexts for access control policies; (2) sets of sensors that can detect those contexts; and (3) evaluations of the security, privacy, and usability of detecting those contexts with those sensors. This section details our method for applying the framework and analyzing each aspect to create Tables 3.2–3.4.

3.4.1 Desirable contexts

Existing work on context sensing does not fully list the desirable contexts for contextual access control in homes. For example, some work focuses on non-security domains, such as sensing contexts for healthcare [83], activity recognition [84, 85], or indoor tracking [86, 87, 88, 89]. Other work focuses on device-level contexts (i.e., device states) [63, 24, 90, 6], but does not consider contextual access control.

To overcome these challenges, we first identified a list of contexts mentioned in the most closely related work on contextual access control in homes [8, 50, 11]. We then analyzed the user study data from He et al. [8]. We manually clustered participant responses through open coding. We added to our list contexts mentioned at least five times or that are related to identity (thus naturally relating to access control). Tables 3.2–3.4 list the final set of desirable contexts in the leftmost column. The "user" in the leftmost column refers to the initiator of the action who uses a device that is owned by the "owner."

3.4.2 Sensing Mechanisms

Extensive prior work proposes technologies to sense identity or contexts in physical spaces. It is hard for a smart home designer to navigate this work and determine the appropriate sensor based on its security, privacy, and usability trade-offs. For example, to track a person's location in the home, researchers have used cameras [87], CSI (Channel State Information) from WiFi signals [89], visible light channels [91], and more. Direct mappings between contexts and precise sensors are not straightforward. Generally, a physical *sensor* is used to sense some characteristic (which we term an *indicator*) that relates to that context. For example, if age is the relevant context, one might use a person's gait, voice, or facial characteristics as physical indicators of age. These indicators can be sensed with cameras, microphones, and more.

For each context, we identified potential indicators and associated sensors by surveying the sensing literature, searching for relevant industry products, and asking experts from the sensing community for methods they had encountered in their field. Our final set of sensors (see Tables 3.2–3.4) includes both research prototypes and mature products. The *example* column of Tables 3.2–3.4 lists the examples of research prototypes or commercial products we consider for each type of sensor.

To find and evaluate research prototypes, we systematically reviewed the last ten years of proceedings of top conferences in sensing systems (SenSys, MobiSys, and MobiCom), ubiquitous computing (UbiComp/IMWUT), and human-computer interaction (CHI and UIST) in the ACM Digital Library. We first filtered the search results based on keywords ("sensing" in the abstract and "home" in the paper), which yielded 716 papers. We then manually inspected each paper to determine its relevance. We used the paper's title to determine potential relevance, which led to 127 papers remaining. We then read each of these papers to determine its actual relevance. We further excluded papers if (i) they were not related to sensing in homes, but rather applications like VR/AR, smart cities, or health; (ii) they did not focus on sensing a specific context, but rather on refining sensing techniques through improved processing algorithms or machine learning techniques; or (iii) we could not directly map the paper to any of the desirable contexts we identified. The final 36 papers are listed in Table 3.2- 3.4, and we extracted the indicators of the contexts from the corresponding papers. If we did not find prototypes in this body of literature for an indicator, we looked to related top-tier conferences, such as CVPR.

To augment this initial list with more mature and commercially viable methods, we first consulted experts in the sensing community to identify classic papers for types of sensors that are now commonly used. To cover methods used in commercial products, we then searched for sensors of each indicator (as collected from research papers above) on Amazon. If we had not found any indicators at that point for a context, we searched for sensors related to that context and then included the indicators they used. This process led to our final set of 94 pairs of a context that is desirable to sense for access control in the home and a type of sensor (research prototype or commercial product) that identifies that context.

The steps described above survey, but do not systematize, this work. For systematization, we applied our framework to analyze the security, privacy, and usability of using that sensor to detect that context. To understand how the sensing method worked, we read the relevant research papers for prototypes and any user manuals, technical specifications, and white papers we could find for commercial products. We list the detailed criteria we use for this systematization below and in Section 3.3.

3.4.3 Security

Attacks, listed in Section 3.2, target particular types of sensors. To perform replay attacks, one must be able to record and then play back the relevant data, a situation that mostly applies to microphones and cameras. Attacks on sensor hardware target sensors' physical properties and are thus relevant to microphones, MEMS sensors, and more. We used past literature to decide whether the type of sensor used by the sensing method is vulnerable or not.

Some attacks (e.g., physical DoS attacks) are less studied and some sensors (e.g., motion sensors) are less often targeted. In these cases, we studied the sensor's basic principles from papers, product manuals, and white papers, and we discussed among our team whether it might be susceptible to each attack. For example, passive infrared (PIR) motion sensors detect motion based on changes in their view in infrared. Infrared radiation struggles to travel through paper, glass, and thermal blankets, which makes occlusion possible. We acknowledge that some products may adopt anti-tampering techniques not specified in the manual or technical specifications. Our judgments reflect contemplation, rather than lab testing. Tables 3.2-3.4 thus outline expected and potential attacks.

3.4.4 Privacy

We evaluated sensors' privacy implications as follows. We identified the data required by each sensor based on its description in its paper or manual. Examples include audio for microphones, air for smoke detectors, and phone packets for CUPID [92], a WiFi-based indoor localization system. We then identified overprivileged data collection by subtracting the information needed to determine the context from what could reasonably be inferred from the required data. We used the guideline in Section 3.3 to label overprivileged data in Tables 3.2-3.4. For example, Touch ID [93] requires fingerprints. This might suggest overprivilege because a fingerprint is personally identifiable. However, since it is used to detect the user's identity, we do not consider its data collection overprivileged.

Next, we determined the data storage location and retention time required for reasonable performance. For storage location, we examined the algorithms needed to process the data for the sensor. If the sensor required a large amount of longitudinal data or algorithms that could not be computed locally (such as Gaussian models), we labeled the sensor as requiring *cloud* storage. Otherwise, we labeled it as *local*. For example, we consider local storage sufficient for sensors that use SVM classifiers and require only highly limited longitudinal data.

If data did not have to be stored for more than one access, we labeled it *transient*. If any data did, then we labeled it *persistent*. For example, smoke detectors have transient data retention because they do not need to store historical air data to detect future smoke. In contrast, fingerprint readers that verify identity do need to store representations (templates) of the fingerprint to perform future matching algorithms.

3.4.5 Limitations

Due to a lack of access to many of the products and prototypes in our evaluation, the ratings we give are based on team discussion and contemplation. To the best of our ability, we tried to make the criteria as concrete as possible and to review papers and specifications with care. However, some cells in Tables 3.2–3.4 could be subjective and debated by researchers with different assumptions and access to different information. As such, we intend Tables 3.2–3.4 to reflect an initial attempt of applying our framework and distilling the pros and cons of each sensor in each context. We intend these tables as a living document that evolves with community effort and robust online debate.

3.5 Insights From Applying the Framework

We present key findings from applying our framework (Section 3.3) to sensors that support commonly desired contextual access-control policies in smart homes. Tables 3.2–3.4 summarize each sensor's pros and cons in security, privacy, and usability regarding detecting a given context.

3.5.1 Robustness to Attacks

Most sensors are vulnerable to physical DoS. Of the 94 context-sensor pairs evaluated, 64 (68.1%) are vulnerable to physical DoS attacks. Vision-, audio-, heat-, and EM-wave-based sensors (radar, WiFi, radio) can easily be blocked or jammed even by those with no technical background. Vision and heat-based sensors' line of sight can be blocked. Playing loud music floods audio sensors. Energy-absorbent materials can be placed near transmitters

				Security						Privacy				Usability						
Contexts	**Indicators**	**Sensor**	**Example**	*Error*	*Replay Attacks*	*Adversarial Examples*	*Physical DoS*	*Sensor Hardware Attacks*	*Spoofing*	*Required Data*	*Overprivileged Data*	*Data Storage*	*Retention Time*	*Wide Availability*	*Initial Set-up / Removal*	*Registration*	*Retraining*	*Reusability*	*Device Dependency*	*Limitations*
User's identity	Voice	Microphone, inertial sensors	[69]	0.1%					!	A,Bm	👎	🏠	○	👎	👍	👍	👍	👎	👎	👎
		Microphone-only	[94]	5-6%						A, C, M	👎	🏠	●	👎	👍	👎	👍	👍	👎	👎
			[41]†	–	!	!	!	!	!	A	👎	☁	○	👍	👍	✋	✋	👍	👍	👎
	Breathing patterns	Microphone	[95]	0.4-2%	?	!				A	👎	🏠	●	👍	👍	✋	👎	👍	👍	👎
	Facial features	Camera	[81]†	Variable	!	!	!			V	👎	☁	●	👍	👍	✋	✋	👍	👍	👍
		Depth camera	[96]†	<0.001%		!	!			P'	👎	🏠	●	👍	👍	✋	✋	✋	👍	👎
		Infrared (IR) camera	[97]†	<0.001%			!			P'	👎	🏠	●	👍	👍	✋	✋	✋	👍	👍
		Camera, inertial, light sensors	[98]	4.7%			!			V, C, E	👎	☁	●	👍	👍	👍	👍	👎	👍	👍
	Eye features	Iris scanner	[99]†	–	!		!			P'	👎	🏠	●	👎	–	✋	👍	👎	👍	👎
	Fingerprint	Fingerprint sensor	[93]†	0.002%	?					F	👍	🏠	●	👎	👍	✋	👍	👎	👍	👎
		Microphone	[100]	2-16%			!	!		A	👎	🏠	●	👍	👍	✋	–	👍	👍	👎
	Body shape	Radar (RF) sensor	[86]	10-21%			!			B	✋	🏠	●	👎	👍	👎	✋	✋	👍	👎
	Bioimpedance	Bioimpedance sensor	[101]	2%						El	👍	🏠	●	👎	👍	✋	👍	👎	👎	👎
			[102]	11-21%						El	👍	🏠	●	👎	✋	👎	–	👎	👍	👎
	Cardiac motion	Radar sensor	[103]	1.39%			!			Bm	✋	🏠	●	👎	👍	✋	👍	👎	👍	👎
		Camera	[104]	1.4-4.5%			!	!		Bm	👎	🏠	●	👍	👍	✋	👍	👍	👍	👎
	Hand gestures	IMU sensors	[105]	10-36.2%			!	!		M	👍	🏠	●	👎	👍	✋	👍	👎	👍	👍
	Gait properties	Vibration sensor	[106]	10%			!			G	✋	🏠	●	👍	👍	✋	–	✋	👎	👎
		Load cells	[88]	7%	?					G	👍	🏠	●	👎	👎	👎	👍	👎	👍	👎
		Pressure sensors	[80]	7.7%	?					G	✋	🏠	●	👎	👎	👎	✋	👎	👍	👎
		Camera	[107]	6.25%	?		!		!	V	👎	🏠	●	👍	👍	👍	✋	👍	👍	👍
		Microphone, WiFi TX & RX	[52]	8%-28%			!			C, A	👎	🏠	●	✋	👍	👎	✋	✋	👍	👎
		Photointerrupters	[108]	1%			!			G	👍	🏠	●	👎	👎	👎	✋	👎	👍	👎
Owner / guest	Identity	*Similar to "Identity" above*		*Similar to "Identity" above*																
User's age	Voice	Microphone	[109]		!	!	!	!	!	A	👎	☁	○	👍	👍	👍	👍	👍	👍	👎
	Facial features	Camera	[110]	6.01 - 6.08 yr.	!	!	!			P	👎	☁	○	👍	👍	👍	👍	👍	👍	👍
			[111]	4.83 - 6.28 yr.	!	!	!			P	👎	☁	○	👍	👍	👍	👍	👍	👍	👍
			[112]	2.514 - 3.086 yr.	!	!	!			P	👎	☁	○	👍	👍	👍	👍	👍	👍	👍
			[113]	22.24 - 9.07%	?	!	!			V	👎	☁	○	👍	👍	👍	👍	👍	👍	👎

Note: In the "Example" column, † denotes commercial sensors or systems.

Table 3.2: An example application of our framework to sensors and contexts identified in our review of the literature and current sensing products. 36 of these sensors come from the academic literature, while the rest are commercial products, denoted with a † in the "Example" column. We mapped the sensors to contexts they are able to detect for the purpose of an access-control policy allowing or denying usage. The "Error" column contains reported values from the cited example sensors. Other columns reflect our best judgment, which was informed by the cited works when related information was reported. !/?/(blank) = Easy/Hard/Impossible, 👍/✋/👎 = Good/Adequate/Poor, 🏠/☁ = Local/Cloud, ○/● = Transient/Persistent data retention, – = Not found. For *Required Data*, **A** = Audio, **B** = Body shape, **Bm** = Body movement, **C** = CSI, **E** = Environment, **El** = Electrical properties of body, **F** = Fingerprint, **G** = Gait, **L/L'** = Geo/Indoor location, **M** = Movement, **P/P'** = Photo/Infrared photo, **D** = Device info, **V/V'** = Video/Infrared video, **T** = Temperature, **O** = Orientation, **Fp** = Floor plan. The rows of this table continue in Table 3.3.

				Security						**Privacy**				**Usability**						
Contexts	**Indicators**	**Sensor**	**Example**	*Error*	*Replay Attacks*	*Adversarial Examples*	*Physical DoS*	*Sensor Hardware Attacks*	*Spoofing*	*Required Data*	*Overprivileged Data*	*Data Storage*	*Retention Time*	*Wide Availability*	*Initial Set-up / Removal*	*Registration*	*Retraining*	*Reusability*	*Device Dependency*	*Limitations*
Emergency in the home	Fire	Smoke detector	[114]†	Variable			!		!	E	👍	🏠	○	👍	👎	👍	👍	👎	👍	👍
			[115]†	Variable			!		!	E	👍	🏠	○	👍	👍	👍	👍	👎	👍	👍
		IR Camera	[116]	Variable			!			V’	👎	☁	○	👎	👍	👍	👍	✋	👍	👍
		IR/UV detector	[117]†	Variable			!			E	👍	🏠	○	👎	👎	👍	👍	👍	👍	👍
	Toxic gas	Combustible gas detector	[118]†	Variable						E	👍	🏠	○	👍	👍	👍	👍	👎	👍	👍
		Carbon monoxide detector	[119]†	Variable						E	👍	🏠	○	👍	👎	👍	👍	👎	👍	👍
	Robbery	Camera	[81]†	Low	!	!	!			V	👎	☁	●	👍	👍	✋	✋	👍	👍	👍
		Glassbreak sensor	[82]†	Variable	!		!	!	!	A	👎	🏠	○	✋	👍	👍	👍	👍	👍	👍
User in same *house* as the device	Tag presence	Bluetooth Low Energy (BLE) signal sensor	[120]†	–	!		?		!	L’	✋	🏠	○	✋	👍	✋	👍	✋	👎	👍
		RF/Ultrasonic sensors	[121]	–			!		!	L’	✋	🏠	○	👎	👍	✋	👍	✋	👎	👍
		RFID	[122]	–			!			L’	👍	🏠	●	👍	👍	👍	👍	👎	👍	👍
		WiFi TX & RX	[94]	10%			!		!	A, C, M	👎	🏠	●	👎	👍	👎	👍	👍	👎	👍
	Movement	WiFi TX & RX	[57]	0.5m - 1.1m			!			C	👍	🏠	○	✋	👎	👍	–	👍	👍	👎
			[123]	1.84m			!			C, Fp	👍	🏠	○	✋	👎	✋	–	👍	👍	👎
			[124]	4%			!			C	✋	🏠	●	👍	👍	✋	✋	👍	👍	👎
	Trajectory	Inertial sensors in phones	[125]	1.5 - 2m	!			!		G, M	👎	🏠	●	👍	👍	👎	👍	👍	✋	👍
User in same *room* as the device	Tag presence	BLE signal sensor	[120]†	–	!		?		!	L’	✋	🏠	○	✋	👍	✋	👍	✋	👎	👍
		BLE, IMU sensors	[126]	2.4 - 14.7%			!	!		M, T, O	👍	☁	●	👍	👍	👎	👎	✋	👍	👍
		RF Techniques	[121]	–			!		!	L’	✋	🏠	○	👎	👍	✋	👍	✋	👎	👍
			[127]	0.06%			!			D	👍	🏠	○	👎	👎	👍	👍	✋	👍	👍
		IR tags	[128]	Variable	!		!		!	L’	👎	🏠	○	👎	👎	👍	👍	👎	👎	👍
		Ultrasound TX & RX	[129]	0.1m	!					L’	👎	🏠	●	👍	👎	👍	–	👍	👎	👎
			[130]	3cm	!					L’	✋	🏠	○	👎	✋	👍	👍	👎	👎	👍
		Capacitive NFC	[131]	–						L’	✋	🏠	○	👎	👎	👍	👍	👎	👍	👍
		Visible Light Channel	[132]	5.9cm			!			L’	✋	🏠	○	👎	✋	👍	👍	👍	👎	👍
	Movement	WiFi TX & RX	[57]	0.5 - 1.1m			!			C	👍	🏠	○	✋	👎	👍	–	👍	👍	👎
			[123]	1.84m			!			C, Fp	👍	🏠	○	✋	👎	✋	–	👍	👍	👎
			[124]	4%			!			C	✋	🏠	●	👍	👍	✋	✋	👍	👍	👎
		Motion sensor	[122]	0.5 - 1.1m			!			M	👍	🏠	●	👍	👍	👍	👍	👎	👍	👍
			[133]†	1.84m			!	!		M	👍	🏠	○	👎	👍	👍	👍	✋	👍	👍
	EMI	Voltage sampling	[134]	6%						L’	✋	🏠	○	👎	✋	👍	👍	👎	👎	👍
		Passive magneto-inductive sensors	[135]	6-17.4%						L’	✋	🏠	○	👎	👍	👎	👍	👎	👎	👍
	RF reflection	RF sensor	[56]	81%			!			L’	👎	☁	●	👎	👍	✋	✋	✋	👍	👍
	Electric potential	Electrical potential sensors	[136]	0.16m			!			El	👎	🏠	○	👎	👍	✋	👍	👎	👍	👍
	Location semantic	WiFi, microphone, IMU sensors, Barometer	[127]	0.63-0.78			!			L’	✋	🏠	○	👍	👍	👍	👍	👍	👎	👍
	Hand gestures	IMU sensor	[105]	10 - 15%				!		L’	✋	🏠	●	👍	👍	👎	–	✋	👎	👎
	Water pressure	Pressure sensor	[137]	17.3 - 29.9%						L’	✋	☁	●	👎	👎	👎	–	👎	👍	👍
Owner away or not	Location	GPS	[138]†	Variable			!		!	L	👎	☁	●	👍	👍	👍	👍	👍	👎	👎
Adult nearby	Age	*Similar to “Age” above*		*Similar to “Age” above*																

Note: In the “Example” column, † denotes commercial sensors or systems.

Table 3.3: A continuation of the rows of Table 3.2, which is an example application of our framework to the sensors and their associated target contexts. The abbreviations used are the same as defined in Table 3.2’s caption.

				Security						Privacy				Usability						
Contexts	**Indicators**	**Sensor**	**Example**	*Error*	*Replay Attacks*	*Adversarial Examples*	*Physical DoS*	*Sensor Hardware Attacks*	*Spoofing*	*Required Data*	*Overprivileged Data*	*Data Storage*	*Retention Time*	*Wide Availability*	*Initial Set-up / Removal*	*Registration*	*Retraining*	*Reusability*	*Device Dependency*	*Limitations*
No one nearby	WiFi signals	WiFi TX & RX	[124]	96% (TPR)			!			C	👍	🏠	○	👍	👍	🫳	🫳	👍	👍	👎
			[139]	Low	!				!	D	🫳	🏠	●	🫳	👍	👍	👍	🫳	👎	👎
	Presence	RF sensor	[86]	High	?		!			B	🫳	🏠	●	👎	👍	👎	🫳	🫳	👍	👎
		Camera with IR	[140]†	Variable			!			V'	👎	☁	●	👍	👍	👍	👍	👍	👍	👍
			[141]	Variable						V'	👎	☁	●	👎	👎	🫳	👎	🫳	👍	👍
		Load cells	[88]	7%						G	👍	🏠	●	👎	👎	👎	👍	👎	👍	👎
		Pressure sensors	[80]	7.7%						G	🫳	🏠	●	👎	👎	👎	👍	👎	👍	👎
		Ultrasonic sensors	[142]	10%			!			B	🫳	🏠	●	👎	👍	👎	👍	🫳	👍	👎
	Movement	Motion sensor	[143]†	Variable			!			M	👍	🏠	○	👎	👍	👍	👍	🫳	👍	👍
			[144]†	Variable			!	!		M	👍	🏠	○	👎	👎	👍	👍	👎	👍	👍
			[145]†	Variable			!	!		M	👍	🏠	○	🫳	🫳	👍	👍	🫳	👍	👍
	Footsteps	Microphones		Variable	?			!	!	A	👎	🏠	●	👍	👍	👍	👍	👍	👍	👍
		Similar to "Gait" above		*Similar to "Gait" above*																
	CO$_2$	Nondispersive Infrared (NDIR) CO$_2$ sensors	[146]†	Variable						E	🫳	🏠	○	🫳	👍	👍	🫳	👎	👍	👍
	Body heat	Infrared sensors	[147]†	Variable			!			V'	👎	🏠	○	👎	👍	👍	👍	🫳	👍	👍
People asleep nearby	Movement	Inertial sensors	[148]†	Variable				!		M	🫳	🏠	○	🫳	👍	🫳	👍	👍	👎	👍
			[149]†	Variable				!		M	🫳	🏠	○	👍	👍	🫳	👍	👍	👎	👍
		Similar to "Motion sensors" above		*Similar to "Motion sensors" above*																
		Radar sensor	[150]	89.6% (recall)			!			M	👎	🏠	○	👎	👍	👍	👍	👎	👍	👍
People present in same *house* as the user	Location	GPS	[138]†	Variable			!		!	L	👎	☁	●	👍	👍	👍	👍	👍	👎	👎
	Movement	Static electrical field	[151]	1.88%			!			E	🫳	🏠	●	👎	👍	👎	–	👎	👎	👍
		RF sensors	[55]	Low			!			M	👎	☁	●	👎	👍	👎	👍	🫳	👎	👍
	Tag presence	RF/Ultrasonic sensors	[121]	–			!		!	L'	🫳	🏠	○	👎	👍	👍	👍	🫳	👎	👍
		BLE signal sensor	[120]†	–	!		?		!	L'	🫳	🏠	○	🫳	👍	🫳	👍	🫳	👎	👍
People present in same *room* as the user	WiFi signals	WiFi TX & RX	[152]	Variable			!			L'	👎	🏠	○	👍	🫳	🫳	–	👍	👍	👍
			[92]	1.8m	!		!			C	🫳	🏠	●	🫳	👍	🫳	👍	🫳	👎	👍
	RF reflection	RF/Ultrasonic sensors	[56]	19%			!			L'	👎	☁	●	👎	👍	🫳	🫳	🫳	👍	👍
	Sound (chat)	RF/Ultrasonic sensors	[94]	26%			!	!		L'	👎	🏠	●	👎	👍	👎	👍	👍	👎	👎
	Doorway activity	RF/Ultrasonic sensors	[142]	10%			!			B	🫳	🏠	●	👎	👍	👍	👍	🫳	👍	👎
	BLE signals	BLE signal sensor	[120]†	–	!		?		!	L'	🫳	🏠	○	🫳	👍	👍	👍	🫳	👎	👍

Note: In the "Example" column, † denotes commercial sensors or systems.

Table 3.4: A continuation of the rows of Table 3.3, which is an example application of our framework to the sensors and their associated target contexts. The abbreviations used are the same as defined in Table 3.3's caption.

(e.g., black material near light-based sensors). Through these means of hindering sensor operation, attackers can become invisible to systems with default-allow policies.

Physical DoS is hard to detect because the symptoms can be similar to normal activities. This is very different from network DoS attacks. Monitoring may alleviate the issue, but home occupants are unlikely to perform constant monitoring. A blocked sensor may not be noticed until the attacker has already achieved their goal.

Sensor redundancy can mitigate physical DoS attacks. For example, a room could have a motion sensor, a pressure sensor in the floor, and a microphone to detect whether the room is occupied or not. If access is granted when the room is unoccupied, an attacker wanting access would need to accomplish the difficult task of occluding all three sensors around the same time. By cross-checking the sensors' data streams with each other [53], the system could verify whether the room is unoccupied and determine whether a sensor has been compromised.

Careful policy design is another defense against physical DoS attacks. A system's default policy—whether to *allow* or *deny* access when a condition is met—can impact attack success. For example, a user might specify "my child should not have access to the TV." With a default-allow policy, TV access will be granted unless a child is detected, yet the child can block a sensor to avoid detection. With a default-deny policy, the child cannot rely on physical DoS.

The optimal default policy may vary based on the device or operation. Users may prefer default-allow rules for controlling lights because falsely allowing operation is typically of little consequence, but falsely denying operation causes inconvenience [8]. A sensor's false positive/negative rates also play a role. Smart home designers should help users navigate these nuances through sensible default policies and templates.

Many sensors are susceptible to physical DoS attacks. Mitigations against physical DoS of sensors include redundant sensors of different types and carefully constructed default policies.

Audio- and vision-based sensing is vulnerable to many attacks. Basic audio-based sensing is susceptible to all types of attacks in Tables 3.2-3.4 [19, 71, 72, 20]. Visible-light camera sensing is also susceptible to all of these attacks, except for hardware attacks. For cameras, spoofing can be difficult, but replay attacks with photo or video input are feasible.

Existing defenses for sensing methods are insufficient for access control because they were designed for *authentication* instead. Most prior work on audio- and camera-based sensing lacks security analyses. The few that analyzed security focused on replay and spoofing attacks. Authentication assumes that unrecognized users are unauthorized. Thus, a large body of research has focused on preventing replay and spoofing attacks against audio- and camera-based sensing to avoid attackers from becoming recognized in this regard. A commonly proposed defense is to rely on secondary channels of information on the same device [153] or other devices [62, 53]. For example, 3D cameras (like Face ID on iPhones [96]) analyze depth information to deter simple, photo-based replay attacks. However, in access control, default-allow policies authorize *unrecognized* users, resulting in the possibility of physical DoS attacks. Therefore, for such policies, an attacker can gain access by targeting one information channel (e.g., targeting an image's visual features by presenting a photo) and becoming unrecognizable to the system.

Existing defenses for audio- and camera-based sensing focus on attacks that compromise authentication, not access control. Attackers can exploit the default semantics of access-control policies to gain access, and physical DoS attacks become easier.

Physical adversarial examples can be effective for skilled, external attackers. For sensing methods that rely on machine learning, we noted whether they were susceptible to adversarial examples. Specifically, within the scope of context sensing and our threat model, we consider only physical adversarial examples. The attacker misleads the algorithms by adding physical perturbations to the environment or to themselves, instead of feeding data to the algorithms directly. Recent work has demonstrated the feasibility of such attacks for images [18, 59, 154] and audio [72, 71, 20]. Although some attacks require whitebox access to models, which is unrealistic for commodity smart home devices, blackbox attacks are also possible [154, 155, 156, 157, 158].

Internal attackers are less likely to use physical adversarial examples because they require substantial technical skills and resources to generate and test. Instead, they would use familiarity with the system to launch replay, spoofing, or physical DoS attacks to a similar end. However, if we consider *external opportunistic* attackers (e.g., a group of burglars) who do not have information about the victim, physical adversarial examples can be very effective. In fact, untargeted adversarial examples are strictly easier than targeted attacks. For example, attackers might want to attack face recognition on all security cameras in a neighborhood. In doing so, they can reuse and refine their adversarial examples.

> Internal attackers may prefer replay, spoofing, and physical DoS attacks. Opportunistic external attackers may prefer adversarial examples.

3.5.2 Privacy

Except for cameras, cloud storage is not usually required when sensing contexts. We found that 79.8% ($n = 75$) of the examined sensing techniques do not require data storage on the cloud. Unfortunately, 10 of the 14 methods that use cameras do require cloud processing. Oftentimes, cloud storage is necessary for computationally intensive algorithms or large training datasets required to process video or image data online (e.g., neural networks for facial recognition). Privacy-preserving machine learning may alleviate this need. One approach is to protect the privacy of the training data. In federated learning [159], sensitive data stays local and only gradient updates are sent to the server. Another approach targets the inference stage by running the models locally or on the edge [160, 161]. Companies may prefer cloud storage because they can collect user data. Despite the risk of data exposure, some users may prefer cloud storage if it costs less.

> Few sensing methods, often camera-based ones, require cloud processing. Federated learning or performing ML on the edge could obviate cloud processing.

Cameras/microphones are invasive but currently indispensable, thus necessitating privacy countermeasures. Users perceive age to be an important context for access control [8]. Unfortunately, most existing age-estimation methods rely on cameras or microphones, raising privacy concerns. Until privacy-preserving methods for age detection become possible, users may instead wish to record age while registering their identity during system setup.

Suppose cameras and microphones have to be used. To enhance bystanders' privacy, countermeasures against these sensing methods have been proposed, such as strategically blurring an image or jamming microphones with ultrasonic noise [162, 21, 22]. These proposals improve privacy, but also imperil the access control system, making it more likely to ignore attackers or confuse attackers with benign users. Therefore, detecting contexts with obfuscated sensor data may be another research direction. Raval et al. [163] proposed a utility-aware obfuscation mechanism for smartphone apps, which shows a promising road to privacy-preserving sensing in homes.

> Privacy-invasive sensors may be essential. Privacy protections may weaken the access-control system.

Mismatch between required and collected data. Only 25 of 94 context-sensor pairs (26.6%) do not collect more data than needed to deduce the context. In contrast, 33.0% were *acceptable* and 40.4% were *poor* in our analysis. Most sensing methods marked as *poor* record unnecessary video or audio. Manufacturers typically rely on high-fidelity sensors, such as cameras or microphones, to sense contexts. This also happens when researchers use microphones on voice assistants or smartphones for ultrasonic-based sensing for their *wide availability*. While federated learning or edge computing may mitigate privacy concerns, they may also appear cryptic to the average user. These methods may therefore fail to alleviate user concerns about sensors inadvertently collecting invasive data. Future work should investigate effective means of communicating to users privacy considerations, such as using privacy labels [164] or visual indicators [165].

Competing interests between multiple stakeholders—manufacturers, researchers, designers, users—also contribute to this mismatch between the data required and the data collected. The designer might only want to know which room the user is occupying, but manufacturers and UbiComp researchers likely would want to collect information about the activity of the user in that room. Obtaining this extra knowledge enables the latter two parties to design and provide technology benefiting users in other aspects of their daily life. For the benefit of smart home owners and users, smart home systems and sensors should offer the ability to prioritize utility or privacy.

> Most sensors collect more data than needed. User awareness and control of data collection is critical.

3.5.3 Access, Deployment, and Acceptability

Many sensing methods for authentication are not inclusive. Research in sensing and access control is generally not inclusive to the elderly and groups with various disabilities. For example, the gait-sensing literature mostly does not consider people with walking disabilities. For inclusivity, contextual access-control systems must offer an array of sensors that allow *every* individual to authenticate an identity or person-specific context.

3.6 Conclusion

Contextual access control in homes is desirable, yet mostly unsupported. To bridge this gap, sensors can be used to detect contexts. However, they must defend against both expert and non-expert adversaries while respecting user privacy and usability. We proposed both a new adversarial model for context sensing in homes and a decision framework for evaluating potential sensors in terms of security, privacy, and usability. We applied this framework to common sensors through literature systematization, finding important trade-offs. We have made our framework and evaluations accessible in a public GitHub repository to facilitate updates and public discussion.

CHAPTER 4

AUTOMATICALLY GENERATING GENERALIZABLE NETWORK ALLOWLISTS FOR HOME IOT DEVICES

Network attacks is another security shortcomings that home IoT devices have suffered from [7]. Attackers have exploited vulnerabilities in home IoT devices' software, protocols, and default settings to take control of devices [166] for purposes including creating botnets like Mirai [167] and Hajime [168]. Contributing to these security issues is the wide variety of (often inexperienced) manufacturers creating home IoT devices, the difficulties of deploying software patches to devices that may not have screens or traditional user interfaces, and the lack of standardization [5].

Rather than relying on potentially unresponsive vendors to patch devices, an appealing solution is for a household to monitor the network traffic of all of its home IoT devices, applying broad security policies designed to disallow problematic network behaviors, such as potential distributed denial of service (*DDoS*) attacks or the exfiltration of data about the home to potentially illegitimate endpoints. From the security perspective, an even more attractive approach would be to employ *allowlists*, which instead enumerate the hosts that can be contacted and block traffic to all other hosts. While allowlist-based approaches have a much smaller attack surface, they are infrequently used in practice because enumerating the destinations that general-purpose computing devices should be able to contact is typically intractable. On the other hand, many home IoT devices typically have a highly limited set of actions and behaviors. Intuition thus suggests that allowlists may be practical for securing home IoT devices at the network level. In addition, allowlists are more challenging to create than blocklists, because they have to be complete to work for all the devices of a product, especially with the widespread usage of load balancing servers. Blocklists, on the contrary, won't affect a device's functionalities if is incomplete.

Therefore, in this proposal, we plan to inspect the possibility of creating and deploying allowlists through crowd-sourcing. We have access to a dataset, the IoT Inspector dataset, that collects aggregated network traffic from real home IoT devices in the wild. With the dataset, we can better understand how different home IoT products work in general. Unlike prior work, the crowd-sourcing dataset gives us a more complete view about how a home IoT product behave in general, what factors may change their behaviors. With such understanding, we have the opportunity to build allowlists that can work for a home IoT product instead of one particular device. It eases the burden of users who previously have to measure their own home IoT devices' network traffic to create an allowlist or a blocklist.

4.1 Project Plans

There are many ways an allowlist can be automatically created. For example, the hosts in the allowlist can take different representations (e.g., IP addresses, hostnames, or domains). Without careful inspection, it is hard to tell how an allowlist should manifest, especially when the IoT ecosystem is highly heterogeneous.

Therefore, the first step in the project is to understand how different design aspects will affect the traffic. Due to the crowd-sourcing nature of the IoT Inspector dataset, many network features are not available due to privacy and ethics consideration, which left us with the following features:

- **Host Representation:** The IoT Inspector dataset contains information about the IP address, host name, and domain name. These representations can all be used for allowlist creation.

- **Regional Information:** The IoT Inspector dataset doesn't collect location information, but keeps timezone information, which can be used as a proxy to geo-location. We wonder if a region-specific allowlist would work better in some circumstances.

- **Sample Size:** To make the solution practical, it is important to know what is the required sample size to create a generally working allowlist.

- **Threshold:** As any crowd-sourcing dataset, it is always possible that some of devices contained in the dataset is already compromised. An attacker can even actively poison the dataset. Therefore, to admit a host to the allowlist, the host must be contacted by N devices in the given sample. The N here is the threshold that one can arbitrarily choose. We would like to understand how different N may affect the allowlist and the allowed traffic.

In addition to creating allowlist from and applying them to the same product, we also would like to investigate if a product's allowlist can be generalized to other products that are made by the same vendor or of the same type. If it could, then it can potentially be used on products that are not included in the dataset, as no dataset can include every single home IoT product in the world.

We plan to test the automatically generated allowlist on both the dataset itself and the real-world devices. The former will give us a more broad evaluation to see how much traffic is blocked due to the allowlist and how the result may change because of the allowlist design. The real-world experiment is designed to see how a real-world device would react to the allowlist, such as whether it will stop working, or, more specifically, when and why it will stop working. It will give us a better understanding about the real impact the allowlist may bring to the devices.

CHAPTER 5

EXPECTED TIMELINE

The main remaining piece about the thesis is the allowlist creation for protecting the network part in our home IoT system model (Chapter 4). Works on user & software (Chapter 2) and environment & hardware part (Chapter 3) have already been done and published [8, 25].

The timeline of finishing the project is as follows.

Remaining Milestones	Deadline
○ Analysis about the IoT Inspector dataset, including host representation, regional impact, etc.	April 2022
○ Allowlist creation algorithm, with consideration of host representation, regions, sample size, and thresholds	May 2022
○ Allowlist evaluation in the real world	June 2022
○ Finishing the thesis and complete the defense	July 2022

REFERENCES

[1] S. Mattu and K. Hill. The house that spied on me, 2018. `https://gizmodo.com/the-house-that-spied-on-me-1822429852`.

[2] Eric Zeng, Shrirang Mare, and Franziska Roesner. End User Security and Privacy Concerns with Smart Homes. In *Proc. SOUPS*, 2017.

[3] Florian Schaub, Rebecca Balebako, Adam L. Durity, and Lorrie Faith Cranor. A Design Space for Effective Privacy Notices. In *Proc. SOUPS*, 2015.

[4] Jingjing Ren, Daniel J. Dubois, David R. Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. Information exposure from consumer iot devices: A multidimensional, network-informed measurement approach. In *Proc. IMC*, 2019.

[5] Nan Zhang, Soteris Demetriou, Xianghang Mi, Wenrui Diao, Kan Yuan, Peiyuan Zong, Feng Qian, XiaoFeng Wang, Kai Chen, Yuan Tian, Carl A. Gunter, Kehuan Zhang, Patrick Tague, and Yue-Hsun Lin. Understanding IoT Security Through the Data Crystal Ball: Where We Are Now and Where We Are Going to Be. *CoRR*, abs/1703.09809, 2017.

[6] Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal, and Chenren Xu. Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things. In *Proc. HotNets*, 2015.

[7] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. Sok: Security evaluation of home-based iot deployments. In *Proc. IEEE SP*, 2019.

[8] Weijia He, Maximilian Golla, Roshni Padhi, Jordan Ofek, Markus Dürmuth, Earlence Fernandes, and Blase Ur. Rethinking access control and authentication for the home internet of things (iot). In *Proc. USENIX Security*, 2018.

[9] Stuart Schechter. The User IS the Enemy, and (S)he Keeps Reaching for that Bright Shiny Power Button! In *Proc. HUPS*, 2013.

[10] Blase Ur, Jaeyeon Jung, and Stuart Schechter. Intruders Versus Intrusiveness: Teens' and Parents' Perspectives on Home-entryway Surveillance. In *Proc. UbiComp*, 2014.

[11] Eric Zeng and Franziska Roesner. Understanding and improving security and privacy in multi-user smart homes: A design exploration and in-home user study. In *Proc. USENIX Security Symposium*, 2019.

[12] Vassilios Lekakis, Yunus Basagalar, and Pete Keleher. Don't Trust Your Roommate or Access Control and Replication Protocols in "Home" Environments. In *Proc. HotStorage*, 2012.

[13] A.J. Bernheim Brush, Jaeyeon Jung, Ratul Mahajan, and Frank Martinez. Digital Neighborhood Watch: Investigating the Sharing of Camera Data Amongst Neighbors. In *Proc. CSCW*, 2013.

[14] Christine Geeng and Franziska Roesner. Who's in control? Interactions in multi-user smart homes. In *Proc. CHI*, 2019.

[15] Aaron Tilley. How A Few Words To Apple's Siri Unlocked A Man's Front Door, September 2016. `https://www.forbes.com/sites/aarontilley/2016/09/21/apple-homekit-siri-security`, as of April 27, 2022.

[16] John Patrick Pullen. Amazon Echo Owners Were Pranked by South Park and Their Alexas Will Make Them Laugh for Weeks, September 2017. `http://fortune.com/2017/09/14/watch-south-park-alexa-echo/`, as of April 27, 2022.

[17] Venessa Wong. Burger King's New Ad Will Hijack Your Google Home, April 2017. `https://www.cnbc.com/2017/04/12/`

`burger-kings-new-ad-will-hijack-your-google-home.html`, as of April 27, 2022.

[18] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proc. CVPR*, 2018.

[19] Guoming Zhang, Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, and Wenyuan Xu. DolphinAttack: Inaudible voice commands. In *Proc. CCS*, 2017.

[20] Nicholas Carlini and David A. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *Proc. DLS*, 2018.

[21] Hyunwoo Yu, Jaemin Lim, Kiyeon Kim, and Suk-Bok Lee. Pinto: Enabling video privacy for commodity IoT cameras. In *Proc. CCS*, 2018.

[22] Yuxin Chen, Huiying Li, Shan-Yuan Teng, Steven Nagels, Zhijing Li, Pedro Lopes, Ben Y. Zhao, and Haitao Zheng. Wearable microphone jamming. In *Proc. CHI*, 2020.

[23] Milijana Surbatovich, Jassim Aljuraidan, Lujo Bauer, Anupam Das, and Limin Jia. Some Recipes Can Do More Than Spoil Your Appetite: Analyzing the Security and Privacy Risks of IFTTT Recipes. In *Proc. WWW*, 2017.

[24] Z. Berkay Celik, Gang Tan, and Patrick D. McDaniel. IoTGuard: Dynamic enforcement of security and safety policy in commodity IoT. In *Proc. NDSS*, 2019.

[25] Weijia He, Valerie Zhao, Olivia Morkved, Sabeeka Siddiqui, Earlence Fernandes, Josiah Hester, and Blase Ur. SoK: Context sensing for access control in the adversarial home iot. In *Proc. EuroS&P*, 2021.

[26] Danny Yuxing Huang, Noah J. Apthorpe, Frank Li, Gunes Acar, and Nick Feamster.

Iot inspector: Crowdsourcing labeled network traffic from smart home devices at scale. *IMWUT*, 4(2), 2020.

[27] Samsung. SmartThings: Add a Little Smartness to Your Things, August 2014. https://www.smartthings.com, as of April 27, 2022.

[28] Amazon. Echo, November 2014. https://www.amazon.com/echo, as of April 27, 2022.

[29] Rayoung Yang and Mark W. Newman. Learning from a Learning Thermostat: Lessons for Intelligent Systems for the Home. In *Proc. UbiComp*, 2013.

[30] Belkin. WeMo Home Automation, January 2012. https://www.belkin.com/wemo, as of April 27, 2022.

[31] Philips. Hue, October 2012. https://www.meethue.com, as of April 27, 2022.

[32] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the Mirai Botnet. In *Proc. USENIX Security Symposium*, 2017.

[33] Earlence Fernandes, Amir Rahmati, Kevin Eykholt, and Atul Prakash. Internet of Things Security Research: A Rehash of Old Ideas or New Intellectual Challenges? *IEEE Security & Privacy*, 15(4):79–84, 2017.

[34] Earlence Fernandes, Jaeyeon Jung, and Atul Prakash. Security Analysis of Emerging Smart Home Applications. In *Proc. IEEE SP*, 2016.

[35] Qi Wang, Wajih Ul Hassan, Adam Bates, and Carl Gunter. Fear and Logging in the Internet of Things. In *Proc. NDSS*, 2018.

[36] Joseph Bonneau, Cormac Herley, Paul C. Van Oorschot, and Frank Stajano. The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes. In *Proc. IEEE SP*, 2012.

[37] Tara Matthews, Kathleen O'Leary, Anna Turner, Manya Sleeper, Jill Palzkill Woelfer, Martin Shelton, Cori Manthorne, Elizabeth F. Churchill, and Sunny Consolvo. Stories from Survivors: Privacy & Security Practices when Coping with Intimate Partner Abuse. In *Proc. CHI*, 2017.

[38] Behrang Fouladi and Sahand Ghanoun. Honey, I'm Home!!, Hacking ZWave Home Automation Systems, July 2013. Black Hat USA.

[39] Blase Ur, Jaeyeon Jung, and Stuart Schechter. The Current State of Access Control for Smart Devices in Homes. In *Proc. HUPS*, 2013.

[40] Samsung. SmartThings: Capabilities Reference, January 2018. `https://smartthings.developer.samsung.com/develop/api-ref/capabilities.html`, as of April 27, 2022.

[41] Google. Set up Voice Match on Google Home, October 2017. `https://support.google.com/googlehome/answer/7323910`, as of April 27, 2022.

[42] Mike Prospero. Best Smart Home Gadgets of 2018, January 2018. `https://www.tomsguide.com/us/best-smart-home-gadgets,review-2008.html`, as of April 27, 2022.

[43] Google. Jacquard Powered Smart Jackets, September 2017. `https://atap.google.com/jacquard/`, as of April 27, 2022.

[44] Huan Feng, Kassem Fawaz, and Kang G. Shin. Continuous Authentication for Voice Assistants. In *Proc. MobiCom*, 2017.

[45] Suman Jana, Arvind Narayanan, and Vitaly Shmatikov. A Scanner Darkly: Protecting User Privacy from Perceptual Applications. In *Proc. IEEE SP*, 2013.

[46] Shrirang Mare, Andrés Molina-Markham, Cory Cornelius, Ronald Peterson, and David Kotz. ZEBRA: Zero-Effort Bilateral Recurring Authentication. In *Proc. IEEE SP*, 2014.

[47] Otto Huhta, Swapnil Udar, Mika Juuti, Prakash Shrestha, Nitesh Saxena, and N. Asokan. Pitfalls in Designing Zero-Effort Deauthentication: Opportunistic Human Observation Attacks. In *Proc. NDSS*, 2016.

[48] Matthew Johnson and Frank Stajano. Usability of Security Management: Defining the Permissions of Guests. In *Proc. SPW*, 2006.

[49] Tamara Denning, Tadayoshi Kohno, and Henry M. Levy. Computer Security and the Modern Home. *CACM*, 56(1):94–103, 2013.

[50] Roei Schuster, Vitaly Shmatikov, and Eran Tromer. Situational access control in the Internet of Things. In *Proc. CCS*, 2018.

[51] Jason I. Hong and James A. Landay. An architecture for privacy-sensitive ubiquitous computing. In *Proc. MobiSys*, 2004.

[52] Yuanying Chen, Wei Dong, Yi Gao, Xue Liu, and Tao Gu. Rapid: A multimodal and device-free approach using noise estimation for robust person identification. *IMWUT*, 1(3), 2017.

[53] Simon Birnbach, Simon Eberz, and Ivan Martinovic. Peeves: Physical event verification in smart homes. In *Proc. CCS*, 2019.

[54] Shaunak Mishra, Yasser Shoukry, Nikhil Karamchandani, Suhas N. Diggavi, and Paulo

Tabuada. Secure state estimation against sensor attacks in the presence of noise. *IEEE TCNS*, 4(1):49–59, 2017.

[55] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. Through-wall human pose estimation using radio signals. In *Proc. CVPR*, 2018.

[56] Chen-Yu Hsu, Rumen Hristov, Guang-He Lee, Mingmin Zhao, and Dina Katabi. Enabling identification and behavioral sensing in homes using radio reflections. In *Proc. CHI*, 2019.

[57] Ju Wang, Hongbo Jiang, Jie Xiong, Kyle Jamieson, Xiaojiang Chen, Dingyi Fang, and Binbin Xie. Lifs: Low human-effort, device-free localization with fine-grained subcarrier information. In *Proc. MobiCom*, 2016.

[58] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino. Impact of artificial "gummy" fingers on fingerprint systems. *OSCDT*, 2002.

[59] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proc. CCS*, 2016.

[60] A.J. Bernheim Brush, Bongshin Lee, Ratul Mahajan, Sharad Agarwal, Stefan Saroiu, and Colin Dixon. Home automation in the wild: Challenges and opportunities. In *Proc. CHI*, 2011.

[61] Dan Simmons. BBC fools HSBC voice recognition security system. BBC, May 2017. `https://www.bbc.com/news/technology-39965545`.

[62] Linghan Zhang, Sheng Tan, and Jie Yang. Hearing your voice is not enough: An articulatory gesture based liveness detection for voice authentication. In *Proc. CCS*, 2017.

[63] Yunhan Jack Jia, Qi Alfred Chen, Shiqi Wang, Amir Rahmati, Earlence Fernandes, Z. Morley Mao, and Atul Prakash. ContexloT: Towards Providing Contextual Integrity to Appified IoT Platforms. In *Proc. NDSS*, 2017.

[64] Blase Ur, Elyse McManus, Melwyn Pak Yong Ho, and Michael L. Littman. Practical trigger-action programming in the smart home. In *Proc. CHI*, 2014.

[65] Qi Wang, Pubali Datta, Wei Yang, Si Liu, Adam Bates, and Carl A. Gunter. Charting the atack surface of trigger-action IoT platforms. In *Proc. CCS*, 2019.

[66] Lefan Zhang, Weijia He, Jesse Martinez, Noah Brackenbury, Shan Lu, and Blase Ur. AutoTap: Synthesizing and repairing trigger-action programs using LTL properties. In *Proc. ICSE*, 2019.

[67] Diana Freed, Jackeline Palmer, Diana Minchala, Karen Levy, Thomas Ristenpart, and Nicola Dell. "A stalker's paradise": How intimate partner abusers exploit technology. In *Proc. CHI*, 2018.

[68] Edoardo Maggio. The facial recognition on Samsung's Galaxy Note 8 can be fooled with a photo. Business Insider, 2017. `https://www.businessinsider.com/samsung-galaxy-note-8-facial-recognition-tricked-with-a-photo-2017-9`.

[69] Huan Feng, Kassem Fawaz, and Kang G. Shin. Continuous authentication for voice assistants. In *Proc. MobiCom*, 2017.

[70] Dibya Mukhopadhyay, Maliheh Shirvanian, and Nitesh Saxena. All your voices are belong to us: Stealing voices to fool humans and machines. In *Proc. ESORICS*, 2015.

[71] Hadi Abdullah, Washington Garcia, Christian Peeters, Patrick Traynor, Kevin R. B. Butler, and Joseph Wilson. Practical hidden voice attacks against speech and speaker recognition systems. In *Proc. NDSS*, 2019.

[72] Lea Schönherr, Katharina Kohls, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding. In *Proc. NDSS*, 2019.

[73] Denis Foo Kune, John D. Backes, Shane S. Clark, Daniel B. Kramer, Matthew R. Reynolds, Kevin Fu, Yongdae Kim, and Wenyuan Xu. Ghost talk: Mitigating EMI signal injection attacks against analog sensors. In *Proc. IEEE S&P*, 2013.

[74] Zhijing Li, Zhujun Xiao, Yanzi Zhu, Irene Pattarachanyakul, Ben Y. Zhao, and Haitao Zheng. Adversarial localization against wireless cameras. In *Proc. HotMobile*, 2018.

[75] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In *Proc. EuroS&P*, 2017.

[76] Sarah Underwood. Distinguishing identical twins. CACM, April 2018.

[77] Eun Kyoung Choe, Sunny Consolvo, Jaeyeon Jung, Beverly Harrison, Shwetak N. Patel, and Julie A. Kientz. Investigating receptiveness to sensing and inference in the home using sensor proxies. In *Proc. UbiComp*, 2012.

[78] Ry Crist. Amazon and Google are listening to your voice recordings. Here's what we know about that. CNET, July 2019. `https://www.cnet.com/how-to/amazon-and-google-are-listening-to-your-voice-recordings-heres-what-we-know/`.

[79] Alex Hern. Uber employees 'spied on ex-partners, politicians and Beyoncé', December 2016. `https://www.theguardian.com/technology/2016/dec/13/uber-employees-spying-ex-partners-politicians-beyonce`.

[80] G. Qian, J. Zhang, and A. Kidané. People identification using floor pressure sensing and analysis. *IEEE Sensors Journal*, 10(9):1447–1460, 2010.

[81] NETATMO. Smart indoor camera. https://www.netatmo.com/en-us/security/cam-indoor/specifications.

[82] Honeywell. 5853 Wireless Glassbreak Detector . https://www.security.honeywell.com/product-repository/5853.

[83] Reham Mohamed and Moustafa Youssef. Heartsense: Ubiquitous accurate multi-modal fusion-based heart rate estimation using smartphones. *IMWUT*, 1(3), September 2017.

[84] Yang Zhang, Chouchang Yang, Scott E. Hudson, Chris Harrison, and Alanson P. Sample. Wall++: Room-scale interactive and context-aware sensing. In *Proc. CHI*, 2018.

[85] Gierad Laput, Karan Ahuja, Mayank Goel, and Chris Harrison. Ubicoustics: Plug-and-play acoustic activity recognition. In *Proc. UIST*, 2018.

[86] Avinash Kalyanaraman, Dezhi Hong, Elahe Soltanaghaei, and Kamin Whitehouse. FormaTrack: Tracking people based on body shape. *IMWUT*, 1(3), 2017.

[87] Chi-Jui Wu, Steven Houben, and Nicolai Marquardt. Eaglesense: Tracking people and devices in interactive spaces using real-time top-view depth-sensing. In *Proc. CHI*, 2017.

[88] Robert J. Orr and Gregory D. Abowd. The smart floor: A mechanism for natural user identification and tracking. In *Proc. CHI EA*, 2000.

[89] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. Widar2.0: Passive human tracking with a single Wi-Fi link. In *Proc. MobiSys*, 2018.

[90] Yuan Tian, Nan Zhang, Yueh-Hsun Lin, Xiaofeng Wang, Blase Ur, Xianzheng Guo, and Patrick Tague. SmartAuth: User-Centered Authorization for the Internet of Things. In *Proc. USENIX Security Symposium*, 2017.

[91] Tianxing Li, Qiang Liu, and Xia Zhou. Practical human sensing in the light. In *Proc. MobiSys*, 2016.

[92] Souvik Sen, Dongho Kim, Stephane Laroche, Kyu-Han Kim, and Jeongkeun Lee. Bringing CUPID indoor positioning system to practice. In *Proc. WWW*, 2015.

[93] Apple. About Touch ID advanced security technology, 2017. `https://support.apple.com/en-us/HT204587`.

[94] Nicholas D. Lane, Petko Georgiev, Cecilia Mascolo, and Ying Gao. Zoe: A cloud-less dialog-enabled continuous sensing wearable exploiting heterogeneous computation. In *Proc. MobiSys*, 2015.

[95] Jagmohan Chauhan, Yining Hu, Suranga Seneviratne, Archan Misra, Aruna Seneviratne, and Youngki Lee. BreathPrint: Breathing acoustics-based user authentication. In *Proc. MobiSys*, 2017.

[96] Apple. Use Face ID on your iPhone or iPad Pro. `https://support.apple.com/en-us/HT208109`.

[97] Microsoft. Windows Hello: Discover facial recognition on Windows 10. `https://www.microsoft.com/en-us/windows/windows-hello`.

[98] Shaxun Chen, Amit Pande, and Prasant Mohapatra. Sensor-assisted facial recognition: an enhanced biometric authentication system for smartphones. In *Proc. MobiSys*, 2014.

[99] Samsung. Galaxy S8 — S8+ - Security. `https://www.samsung.com/global/galaxy/galaxy-s8/security/`.

[100] Aditya Singh Rathore, Weijin Zhu, Afee Daiyan, Chenhan Xu, Kun Wang, Feng Lin, Kui Ren, and Wenyao Xu. Sonicprint: A generally adoptable and secure fingerprint biometrics in smart devices. In *Proc. MobiSys*, 2020.

[101] Cory Cornelius, Ronald Peterson, Joseph Skinner, Ryan Halter, and David Kotz. A wearable system that knows who wears it. In *Proc. MobiSys*, 2014.

[102] Munehiko Sato, Rohan S. Puri, Alex Olwal, Yosuke Ushigome, Lukas Franciszkiewicz, Deepak Chandra, Ivan Poupyrev, and Ramesh Raskar. Zensei: Embedded, multi-electrode bioimpedance sensing for implicit, ubiquitous user recognition. In *Proc. CHI*, 2017.

[103] Feng Lin, Chen Song, Yan Zhuang, Wenyao Xu, Changzhi Li, and Kui Ren. Cardiac scan: A non-contact and continuous heart-based user authentication system. In *Proc. MobiCom*, 2017.

[104] Jian Liu, Cong Shi, Yingying Chen, Hongbo Liu, and Marco Gruteser. Cardiocam: Leveraging camera on mobile devices to verify users while their heart is pumping. In *Proc. MobiSys*, 2019.

[105] Juhi Ranjan and Kamin Whitehouse. Object hallmarks: Identifying object users using wearable wrist sensors. In *Proc. UbiComp*, 2015.

[106] Shijia Pan, Tong Yu, Mostafa Mirshekari, Jonathon Fagert, Amelie Bonde, Ole J. Mengshoel, Hae Young Noh, and Pei Zhang. FootprintID: Indoor pedestrian identification through ambient structural vibration sensing. *IMWUT*, 1(3), 2017.

[107] Liang Wang, Tieniu Tan, Huazhong Ning, and Weiming Hu. Silhouette analysis-based gait recognition for human identification. *IEEE TPAMI*, 25(12), 2003.

[108] Jaeseok Yun. User identification using gait patterns on UbiFloorII. *Sensors*, 11(3), 2011.

[109] Mohammad Sedaaghi. A comparative study of gender and age classification in speech signals. *IJEEE*, 5, 03 2009.

[110] Dat Tien Nguyen, So Ra Cho, Tuyen Danh Pham, and Kang Ryoung Park. Human age estimation method robust to camera sensor and/or face movement. *Sensors*, 15(9), 2015.

[111] Yu Zhang and Dit-Yan Yeung. Multi-task warped Gaussian process for personalized age estimation. In *Proc. CVPR*, 2010.

[112] Hongyu Pan, Hu Han, Shiguang Shan, and Xilin Chen. Mean-variance loss for deep age estimation from a face. In *Proc. CVPR*, 2018.

[113] Hamdi Dibeklioglu, Fares Alnajar, Albert Ali Salah, and Theo Gevers. Combining facial dynamics with appearance for age estimation. *IEEE TIP*, 24(6), 2015.

[114] Google Nest. Split-spectrum white paper. Technical report, June 2015.

[115] First Alert. Battery powered photo & ion smoke alarm. `https://images-na.ssl-images-amazon.com/images/I/A1+UJjI+uPL.pdf`.

[116] B. Ugur Töreyin, R. Gokberk Cinbis, Yigithan Dedeoglu, and A. Enis Cetin. Fire detection in infrared video using wavelet analysis. *Opt. Eng.*, 46(6), 2007.

[117] Sierra Monitor Corporation. `https://www.sierramonitor.com/flame-detector-3600-lb`.

[118] Techamor. `https://www.amazon.com/dp/B07BM1XWB8`.

[119] Google. Google Nest Protect. `https://store.google.com/us/product/nest_protect_2nd_gen`.

[120] Apple. iBeacon. `https://developer.apple.com/ibeacon/`.

[121] Nissanka B. Priyantha, Anit Chakraborty, and Hari Balakrishnan. The Cricket location-support system. In *Proc. MobiCom*, 2000.

[122] James Scott, A.J. Bernheim Brush, John Krumm, Brian Meyers, Michael Hazas, Stephen Hodges, and Nicolas Villar. Preheat: Controlling home heating using occupancy prediction. In *Proc. UbiComp*, 2011.

[123] Kazuya Ohara, Takuya Maekawa, Yasue Kishino, Yoshinari Shirai, and Futoshi Naya. Transferring positioning model for device-free passive indoor localization. In *Proc. UbiComp*, 2015.

[124] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures. In *Proc. MobiCom*, 2014.

[125] Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao. A reliable and accurate indoor localization method using phone inertial sensors. In *Proc. Ubicomp*, 2012.

[126] Gabriele Civitarese, Stefano Belfiore, and Claudio Bettini. Let the objects tell what you are doing. In *Proc. UbiComp*, 2016.

[127] Yang Zhang, Yasha Iravantchi, Haojian Jin, Swarun Kumar, and Chris Harrison. Sozu: Self-powered radio tags for building-scale activity sensing. In *Proc. UIST*, 2019.

[128] Roy Want, Andy Hopper, Veronica Falcão, and Jonathan Gibbons. The active badge location system. *ACM TIS*, 10(1):91–102, 1992.

[129] Patrick Lazik and Anthony Rowe. Indoor pseudo-ranging of mobile devices using ultrasonic chirps. In *Proc. SenSys*, 2012.

[130] Mike Addlesee, Rupert Curwen, Steve Hodges, Joe Newman, Pete Steggles, Andy Ward, and Andy Hopper. Implementing a sentient computing system. *Computer*, 34(8):50–56, 2001.

[131] Tobias Grosse-Puppendahl, Sebastian Herber, Raphael Wimmer, Frank Englert, Sebastian Beck, Julian von Wilmsdorff, Reiner Wichert, and Arjan Kuijper. Capacitive near-field communication for ubiquitous interaction and perception. In *Proc. UbiComp*, 2014.

[132] Weizhi Zhang, MI Sakib Chowdhury, and Mohsen Kavehrad. Asynchronous indoor positioning system based on visible light communications. *Opt. Eng.*, 53(4), 2014.

[133] NAPCO. Napco adaptive dual microwave/PIR detector, 30x35 ft. (c-100ste). `https://www.amazon.com/Napco-Adaptive-Microwave-Detector-C-100STE/dp/B0041X47EW`.

[134] Sidhant Gupta, Ke-Yu Chen, Matthew S. Reynolds, and Shwetak N. Patel. Lightwave: Using compact fluorescent lights as sensors. In *Proc. UbiComp*, 2011.

[135] Edward J. Wang, Tien-Jui Lee, Alex Mariakakis, Mayank Goel, Sidhant Gupta, and Shwetak N. Patel. Magnifisense: Inferring device interaction using wrist-worn passive magneto-inductive sensors. In *Proc. UbiComp*, 2015.

[136] Tobias Grosse-Puppendahl, Xavier Dellangnol, Christian Hatzfeld, Biying Fu, Mario Kupnik, Arjan Kuijper, Matthias R. Hastall, James Scott, and Marco Gruteser. Platypus - Indoor localization and identification through sensing electric potential changes in human bodies. In *Proc. MobiSys*, 2016.

[137] Edison Thomaz, Vinay Bettadapura, Gabriel Reyes, Megha Sandesh, Grant Schindler, Thomas Plötz, Gregory D. Abowd, and Irfan Essa. Recognizing water-based activities in the home through infrastructure-mediated sensing. In *Proc. UbiComp*, 2012.

[138] National Coordination Office for Space-Based Positioning, Navigation, and Timing. GPS: The Global Positioning System. `https://www.gps.gov/`.

[139] Bharathan Balaji, Jian Xu, Anthony Nwokafor, Rajesh Gupta, and Yuvraj Agarwal. Sentinel: Occupancy based HVAC actuation using existing WiFi infrastructure within commercial buildings. In *Proc. SenSys*, 2013.

[140] Google Nest Cam. Indoor - tech specs. https://store.google.com/us/product/nest_cam_specs.

[141] Alan Bränzel, Christian Holz, Daniel Hoffmann, Dominik Schmidt, Marius Knaust, Patrick Lühne, René Meusel, Stephan Richter, and Patrick Baudisch. GravitySpace: Tracking users and their poses in a smart room using a pressure-sensing floor. In *Proc. CHI EA*, 2013.

[142] Timothy W. Hnat, Erin Griffiths, Raymond Dawson, and Kamin Whitehouse. Doorjamb: Unobtrusive room-level tracking of people in homes using doorway sensors. In *Proc. SenSys*, 2012.

[143] Samsung. Motion sensor. https://www.lowes.com/pd/Samsung-Motion-Sensor/1000555661.

[144] NAPCO. Napco's adaptive dual microwave/PIR detectors automatically adjust to their environment, minute by minute, for the ultimate false alarm immunity & reliability. https://napcosecurity.com/products/napco-detectors/.

[145] Honeywell. DT906 / DT907. https://www.security.honeywell.com/product-repository/dt906-dt907.

[146] CO2Meter. Tim10 desktop co2, temp. & humidity monitor. https://www.co2meter.com/products/tim10-desktop-co2-temp-humidity-monitor.

[147] GridEye. Infrared Array Sensor Grid-EYE: High Precision Infrared Array Sensor based on Advanced MEMS Technology. https://www.mouser.com/datasheet/2/315/ADI8000C65-1267019.pdf.

[148] Apple. Healthkit. https://developer.apple.com/healthkit/.

[149] Fitbit Inc. How do I track my activity with my Fitbit device? https://help.fitbit.com/articles/en_US/Help_article/1785.

[150] Tauhidur Rahman, Alexander T. Adams, Ruth Vinisha Ravichandran, Mi Zhang, Shwetak N. Patel, Julie A. Kientz, and Tanzeem Choudhury. Dopplesleep: A contactless unobtrusive sleep sensing system using short-range doppler radar. In *Proc. UbiComp*, 2015.

[151] Adiyan Mujibiya and Jun Rekimoto. Mirage: Exploring interaction modalities using off-body static electric field sensing. In *Proc. UIST*, 2013.

[152] Sheng Tan, Linghan Zhang, Zi Wang, and Jie Yang. Multitrack: Multi-user tracking and activity recognition using commodity wifi. In *Proc. CHI*, 2019.

[153] Di Tang, Zhe Zhou, Yinqian Zhang, and Kehuan Zhang. Face Flashing: A secure liveness detection protocol based on light reflections. In *Proc. NDSS*, 2018.

[154] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Physical adversarial examples for object detectors. In *Proc. WOOT*, 2018.

[155] Chun-Chen Tu, Pai-Shun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In *Proc. AAAI*, 2019.

[156] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *Proc. ICLR*, 2019.

[157] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *Proc. ICLR*, 2017.

[158] Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *Proc. USENIX Security*, 2020.

[159] Jakub Konečnỳ, H. Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. arXiv:1610.02527, 2016.

[160] Aditya Kusupati, Manish Singh, Kush Bhatia, Ashish Kumar, Prateek Jain, and Manik Varma. FastGRNN: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network. In *Proc. NeurIPS*, 2018.

[161] Sridhar Gopinath, Nikhil Ghanathe, Vivek Seshadri, and Rahul Sharma. Compiling KB-sized machine learning models to tiny IoT devices. In *Proc. PLDI*, 2019.

[162] Mariella Dimiccoli, Juan Marín, and Edison Thomaz. Mitigating bystander privacy concerns in egocentric activity recognition with deep learning and intentional image degradation. *IMWUT*, 1(4), 2017.

[163] Nisarg Raval, Ashwin Machanavajjhala, and Jerry Pan. Olympus: Sensor privacy through utility aware obfuscation. *PoPETS*, 2019(1):5–25, 2019.

[164] Pardis Emami Naeini, Yuvraj Agarwal, Lorrie Faith Cranor, and Hanan Hibshi. Ask the experts: What should be on an IoT privacy and security label? In *Proc. IEEE S&P*, 2020.

[165] Serge Egelman, Raghudeep Kannavara, and Richard Chow. Is this thing on?: Crowdsourcing privacy indicators for ubiquitous sensing platforms. In *Proc. CHI*, 2015.

[166] Ezra Caltum and Ory Segal. Sshowdown - exploitation of iot devices for launching mass-scale attack campaigns, 2016. https://tinyurl.com/3tbe358x.

[167] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. Understanding the mirai botnet. In *Proc. USENIX Security*, pages 1093–1110, 2017.

[168] Stephen Herwig, Katura Harvey, George Hughey, Richard Roberts, and Dave Levin. Measurement and analysis of hajime, a peer-to-peer iot botnet. In *Proc. NDSS*, 2019.