The Privacy Landscape of Pervasive Computing

This roadmap for privacy research uses two different lenses—one focused on devices and another focused on entities in the privacy ecosystem. The author argues for shifting some of the burden of privacy from users to developers, service providers, governments, and third parties.

Jason Hong
Carnegie Mellon University

It has been more than 25 years since Mark Weiser first introduced us to ubiquitous computing.[1] If we could travel back in time and return with someone from 1991, that person would agree that we've realized Weiser's vision of weaving computation, communication, sensing, and actuation into the physical world. You can now go into any big box store and purchase smartphones, tablets, wearable fitness trackers, webcams, drones, smart thermostats, network-enabled toys, and more.

However, in their initial deployments of ubicomp systems, Weiser and his team also hit upon the issue of privacy, which we're still struggling with today. There is a fundamental tension with all of this rich sensor and log data that ubicomp devices can collect. On the one hand, this data can be used for personal and societal benefit, improving healthcare, sustainability, transportation, education, urban planning, finance, and more. On the other hand, these same technologies introduce many new privacy risks, often at such a fast rate that legal mechanisms and social norms can't keep up.

In a ubiquitously connected world, the costs of collecting, storing, inferring, searching, and sharing data are dramatically lower. The privacy risks range from everyday ones—such as being monitored by overprotective parents, feeling the need to meet undesired social obligations with friends and family members, and receiving overly intrusive marketing—to extreme ones, such as threats to civil liberties by governments and to personal safety by stalkers, muggers, and domestic abusers. Every day, it seems there is some new headline or post describing people's concerns with pervasive computing technologies—whether it be about the potential for abuse, a general unease stemming from the lack of control, or an overall desire for privacy. In some cases, these concerns have led to outright rejection of systems. For example, Pew Internet found that 54 percent of app users chose not to install a smartphone app when they discovered how much personal information the app requested, and 30 percent of app users uninstalled an app after learning that it was collecting personal information that they did not want to share.[2] In short, privacy might be the greatest barrier to creating a ubiquitously connected world.

In 2001, Mahadev Satyanarayanan outlined a broad research agenda for pervasive computing.[3] Here, I dive deeper into his discussion of privacy and present a roadmap specifically for privacy research in pervasive computing. This roadmap is not comprehensive, as privacy is too broad of a sociotechnical issue for any single article to address. Instead, I highlight some challenges and present opportunities our community can leverage to address privacy concerns. I organize these opportunities using two

Published by the IEEE CS ■ 1536-1268/17/$33.00 © 2017 IEEE

different lenses—the first focuses on different tiers of devices, and the second focuses on different entities involved with privacy, including users, developers, service providers, and third parties. I advance the argument that today, too much of the burden of privacy is on users. To achieve a sustainable ecosystem, we need to develop new ways of shifting this burden to other entities.

Understanding Pervasive Devices

Pervasive computing—including its current industry name, the Internet of Things—is often talked about as a single monolithic concept. However, it's more useful to think of it in terms of a three-tier pyramid (see Figure 1). Each tier represents a different device class, based on the device capabilities as well as our relationship to the device. Each tier also poses different kinds of privacy challenges based on the capabilities of the devices in that tier.

At the top of the pyramid are devices with a great deal of computational heft and with rich sensing capabilities, fast networking, long battery life, and high interactivity. These devices are highly personal and are what people typically think of as computers. Examples include laptops, smart glasses, tablets, smartphones, and gaming devices. Each person has only a few of these devices but spends a lot of time with them. Most of these devices have common operating systems, can run third-party software, and are manufactured by large corporations experienced in developing secure software.

In the middle are devices that offer basic interactivity, such as TVs, smart watches, refrigerators, thermostats, electronic whiteboards, cable boxes, and interactive toys. Some of these devices have advanced sensing and computing capabilities, but the key characteristic is that people use them at most a few times a day, and this use requires little attention. There is greater diversity here in terms of manufacturers, operating systems, and software development experience.

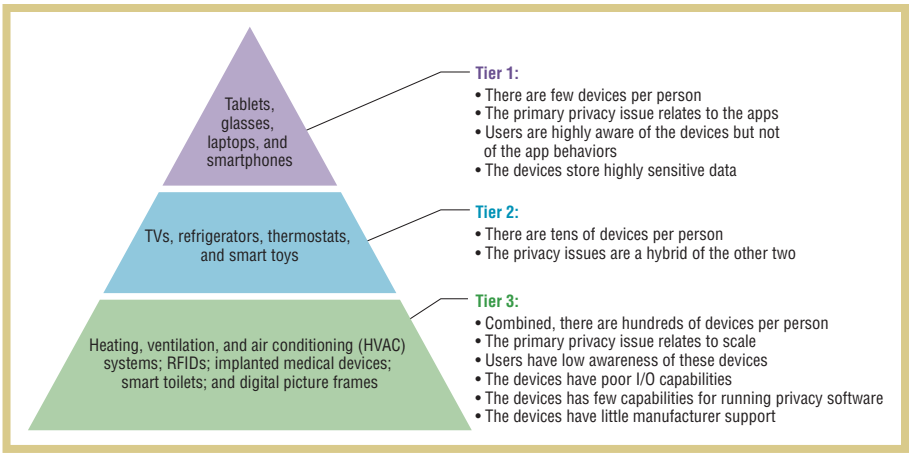


Figure 1. The three tiers of pervasive computing. Each tier contains devices with a different set of characteristics.

At the bottom of the pyramid, there are hundreds of devices per person, and each device lies far in the background of our attention. These might include RFID-enabled ID cards and badges; wearable devices embedded in clothing; heating, ventilation, and air conditioning (HVAC) systems; digital lightbulbs; smart toilets; smart meters; security systems; implanted medical devices; digital picture frames; cheap environmental sensors; and electronic locks. Most of these devices are embedded or situated in homes, buildings, and public places. They have limited computational resources, basic sensing, few (if any) software capabilities, and a wide range of software and operating systems. Many of the manufacturers of devices in this tier also have little experience in developing reliable software and pushing out updates.

Privacy Dimensions

Here, I sketch out seven dimensions of privacy for describing these tiers.

The first dimension, *awareness*, examines how easy it is for people to be aware of both what might be sensed in a given situation as well as what is actually being sensed. The next dimension, *depth of sensing*, indicates the richness of what can be sensed by a given device. *Temporal scale* refers to the scope of time in which a device will be sensing data about a person, while *I/O* describes how much input and output capabilities a device has for interacting with user interfaces related to privacy—for example, configuring policies and viewing log data. *Privacy software* describes how much privacy and security-related software can be run on the device itself. In contrast, *third-party software* describes whether software developed by others can be installed—and run—on the device. The final dimension, *manufacturer support*, relates to how much ongoing support manufacturers give in terms of fixing bugs and upgrading software.

Each of the three tiers has similar kinds of privacy challenges, but the tiers differ primarily in terms of degree and scale. In this article, I focus only on the top and bottom tiers, because they represent opposite sides of the spectrum (see Table 1). The middle tier will be a hybrid of these two sides.

For devices in the top tier, people have high awareness, primarily because most of these devices have stylish form factors and are often readily available for interaction. These devices also have rich sensing capabilities and can sense our physical and social contexts. For temporal scale, we spend a great deal of time with these devices, often carrying them as we move from place to place throughout the day. In terms of

TABLE 1
Privacy characteristics of the top and bottom tiers of pervasive devices.

Privacy dimension	Top-tier devices	Bottom-tier devices
Awareness	High device (but not app-behavior) awareness	Low device awareness
Depth of sensing	Rich sensing capabilities	Limited sensing capabilities
Temporal scale	High—people spend a great deal of time with these devices	Low—although these devices capture a significant amount of data over time, they can't always differentiate between specific individuals
I/O	High-quality I/O	Limited I/O
Privacy software	Privacy-enhancing technologies and security protocols	Little support for privacy-enhancing technologies
Third-party software	Lots of third-party software	Little third-party software
Manufacturer support	Experience in developing reliable software and pushing out software updates	Little experience in developing software or pushing out software updates

I/O, these devices offer text, pointing, speech, and more, and often have high-end displays that can be used to display privacy-related information. For privacy software, these devices have enough computational power to run necessary privacy-enhancing technologies and security protocols. Users can also install third-party software, most likely through app stores. In fact, the community's experience with smartphones suggests that these apps might present the most difficult aspect of privacy for this class of device. Perhaps most importantly, manufacturers of these devices have a lot of experience in developing reliable software and pushing out software updates to fix potential privacy and security software bugs.

Devices in the bottom tier are in sharp contrast to those in the top tier. For awareness, such devices are hard to see and are often part of the physical infrastructure. For sensing, they typically have few sensors and will be severely constrained by smaller batteries, limiting how much they can sense. In terms of temporal scale, most of these devices are embedded in specific places, meaning they can capture a great deal of data over time but might not always be able to differentiate between specific individuals. With respect to I/O, these devices have little or no capabilities, making it difficult to configure and manage privacy. They also have little support for third-party software, typically only running manufacturer-installed software. Lastly, there are hundreds of manufacturers of these low-end devices, most of which have little experience in developing software or pushing out software updates to fix bugs. Many devices will never get software updates because of high developer costs or a lack of user awareness, or because the manufacturer has gone out of business.[4] More importantly, all of these issues will be exacerbated by scale.

Privacy Challenges

Given this context, privacy-related research challenges include the following.

Increasing awareness. One of the most salient issues is improving users' awareness of what data is being gathered, where it is being sent (to Facebook or a cloud service, for example), and how it is being used. For top-tier devices, people often already know that the device is there. As such, the core issue is awareness of what the apps are doing. There is a growing body of research in this area, focusing primarily on smartphones. For bottom-tier devices, there is the same concern about apps, but more important is awareness of the devices themselves, because many of the devices are small and possibly even part of the physical infrastructure.

I pose a research challenge to the community: let's make it so that when a person enters a room, he or she can reliably identify all of the sensors and data flows within 30 seconds. This challenge leads to many questions. Should users be notified through Bluetooth low-energy beacons (similar to the *p*rivacy *aw*areness *s*ystem, PAWS[5]), through commonly located visual displays (such as near the light switches in a room), or LEDs or audio on sensors? What kinds of common interaction languages can be developed to inform people about sensors? Can we develop cost-benefit models that can balance the benefits of awareness with the costs of user attention and annoyance from repeated or uninteresting notifications? One step forward in addressing these challenges is a project I helped with that explored different kinds of audio, visual, and motion feedback to convey to users that devices are actively sensing them (http://signifiers.io).

Improving I/O for privacy information. Another privacy issue relates to input and output. Top-tier devices can support high-quality user interactions with

privacy-related UIs, but bottom-tier devices have poor I/O capabilities. Exacerbating this issue is scale. The sheer number of pervasive computing devices makes what would ordinarily be trivial tasks into significant challenges. Configuring privacy for a single device is tractable. Configuring privacy for hundreds of devices, each of which has a different user interface and different sensing capabilities, is not.

One research thrust here is developing better tools to help configure and manage multiple devices and apps with respect to privacy. Can we create new kinds of network protocols as well as system architectures that make it easy to add, configure, and maintain devices, all in the context of privacy? Can we also help people make good decisions—for example, helping them avoid misconfigurations and make privacy choices with which they're comfortable? Are there ways we can use crowdsourcing or AI techniques to help people make good "trust" decisions and to help expose relevant privacy-related events and potential abuses?

Including basic privacy functionality. A third privacy issue is that the vast majority of low-end devices have minimal capabilities in running privacy and security software. Given scalability issues, the infrastructure should offer basic privacy and security functionality. What kinds of services can facilitate the development and deployment of these bottom-tier devices? For example, some essential features might include access control, network security, and software updates.

Are there also new ways of conceptualizing these features in the context of pervasive computing? Can we use proximity to a person or room to simplify access control (for example, to make it so people can always access basic services and sensors in the room they're currently in)? As another example, can we better help with data-sharing decisions, given that research has shown that who is requesting the data is an important factor?[6,7] Although it seems like a privacy paradox, perhaps can we use big data approaches to model people and places to improve privacy. For example, perhaps we could use sensor data to infer people's relationships with each other, so as to offer useful defaults for sharing.

Fostering an Ecosystem for Privacy

Moving away from the perspective of devices, here I examine privacy from the perspective of the different players involved in pervasive computing deployments.

Today, users primarily shoulder the burden of privacy. Individuals must know a great deal about their devices' capabilities and behaviors and the operating systems and apps they use. Individuals also must know about hidden metadata (such as geotags in photos) and how to configure user interfaces and determine the right settings. Furthermore, devices owned by others or embedded in the physical infrastructure might also be monitoring individuals. In short, individuals must have a great deal of awareness and knowledge as to how to protect themselves, and they must take many affirmative steps to enable such protection.

Instead, I argue that we need to expand privacy research to foster an *ecosystem for privacy*. Using an analogy with spam email, in the early 2000s, people had to spend a great deal of time manually deleting spam from their inbox. However, over time, email service providers started to deploy spam filters, Internet Service Providers started to coordinate in developing blacklists to filter out certain IP addresses, and law enforcement worked with industry to take down botnets and arrest individuals responsible for spam. Although spam is still an ongoing problem, users manually handle far fewer spam emails because of the rest of the ecosystem.

Similarly, can we foster an ecosystem for privacy for pervasive computing, one that can help shift the burden of privacy from being solely on users? How can we, as researchers, empower other entities to help manage and protect privacy for users? Examining the ecosystem for smartphones is instructive here, because smartphones are the most developed form of pervasive computing technologies. Today's smartphone ecosystem can be broken down into the following entities:

- *Developers*, the people who design and create apps.
- *Third-party developers*, the people who create reusable libraries that help other developers.
- *Service providers*, the organizations that deploy apps or offer supporting networked services (such as advertising, analytics, maps, social networking, or cloud storage).
- *App stores*, the organizations that aggregate and distribute apps (today, this would be primarily Google, Apple, Amazon, and Microsoft).
- *OS providers* (today, primarily Google and Apple).
- *Hardware manufacturers*, such as Samsung, LG, Motorola, and Apple.
- *Government agencies and third parties interested in privacy*, including the US Federal Trade Commission and equivalents, Electronic Frontier Foundation, Center for Democracy and Technology, journalists, and independent websites (such as PrivacyGrade.org) that rate the privacy of apps.
- *Users*, the people who download and use the apps.

We will likely see a similar set of players for pervasive computing. Using this framing, I sketch out some opportunities for privacy research in Table 2. Note that I have combined third-party developers and service providers, because these two are often the same entity. I also skip over hardware manufacturers in my discussion, in part because I've already discussed privacy and devices. However, from a business perspective, we should consider leveraging hardware manufacturers, because they often already have

TABLE 2
Different entities in the pervasive computing ecosystem for privacy and opportunities for privacy research.

Entity	Opportunities for privacy research
Developers	Develop verifiable explanations for sensitive data use. Create better reusable components. Weave and enforce privacy properties throughout an app.
Third-party developers and service providers	Present clearer explanations for developers of what their software libraries/services do and why. Offer new ways of building and deploying privacy-sensitive services. Offer new ways of deploying advertisements while minimizing collection of personally identifiable information.
App stores	Develop more effective methods of scanning for potential privacy problems at scale. Simplify summaries of potential privacy issues with apps.
OS and middleware manufacturers	Support software updates on low-end devices. Develop formal or inferred models of "normal" device behavior. Present different levels of programming abstractions for accessing sensitive data.
Hardware manufacturers	Present visual, audio, and physical feedback about sensed data. Align privacy with power consumption and the user experience.
Government and third parties	Create tools for checking privacy-related behaviors of apps. Create tools for monitoring behaviors of service providers. Raise awareness by developing easily understandable summaries of apps.
Users	Gain a better understanding of user concerns about privacy. Create simpler mechanisms for awareness and control. Find better ways of expressing preferences.

a clear business model for selling the devices themselves—rather than an advertising model, which entails collecting data about their users. As such, they might be interested in including privacy-enhancing technologies with their devices, especially if those technologies can also improve other features, such as battery life or the user experience.

Developers

There have been many system architectures and data processing techniques developed for privacy-sensitive pervasive computing systems. However, most of this work has focused on one data type—location.

Furthermore, past research has found that app developers have little knowledge of laws or best practices surrounding data collection.[8] In short, developers still need a lot more help in creating privacy-sensitive systems. Below, I sketch three promising paths for assisting developers.

People care a lot about who is requesting personal data and why,[6,9,10] but there is currently little support for these kinds of explanations. Long privacy policies are burdensome. iOS offers rudimentary support for explaining why sensitive data is being requested, but these explanations are just plain text. What if developers had to specify their purpose from a known set? A major challenge would be to develop an agreed-upon ontology, but if we could do that, these "purposes" could help automatically generate displays of why an app uses your information (such as "this app uses your location data for ads" or "this app uses your contact list for social networking"). Such an ontology could also facilitate the development and use of various tools to check the displayed purpose against the app's actual behaviors, enabling new kinds of context-based access control ("allow microphone only for social media and health apps"). These explanations might be supported through new techniques for annotating source code or augmenting manifest files.

Developers also need a high degree of skill in managing all of the sensitive data they collect. Can we create better reusable components that offer useful privacy properties? To use an analogy, databases offer Atomic, Consistent, Isolated, and Durable (ACID) properties, but a developer doesn't have to know anything about these properties—or about how a database works—to benefit from using databases. As such, one research direction would be enumerating a set of useful and desirable privacy properties, as well as creating tools that either have or can support those properties. Another possible starting point might be to understand what kinds of difficulties developers face when they try implementing privacy features and to look for opportunities for reuse or generalizability.

A related path would be finding better ways of expressing desired properties or policies and then weaving them into an

application directly or having them be enforced system-wide. Examples for privacy might include direct support for privacy in programming languages (such as taint tracking or other kinds of metadata tags on variables), or making it easier to ensure that a set of policies is enforced as the code is compiled.[11]

A key challenge for these lines of research is coming up with a list of desired properties that are meaningful and feasible. For example, in security, researchers often talk about confidentiality, integrity, and availability. At a high level, some desired privacy properties might include ensuring anonymity, minimizing interruptions, avoiding spam, offering data-flow control and feedback, avoiding behavioral advertising, and preventing embarrassing content from being seen by friends and family. Fair Information Practices might also offer some other desired properties, such as notice, consent, recourse, and so on. However, it's currently not clear how many of these properties can be implemented. There are promising advances—and one in particular is *differential privacy*,[12] a principled way of adding noise to sets of data so that analysts can still get useful aggregate results without learning information about specific individuals. However, much like Isacc Asimov's "Three Laws of Robotics," there is still a very wide technical gap between these high-level declarations and the code needed to implement them.

Third-Party Developers and Service Providers

If pervasive computing follows the evolution of smartphone apps, we will see a host of third parties offering useful libraries and services. However, it could be difficult to understand the privacy-related behaviors of these components. For example, in some cases, it's not the developers themselves but rather third-party libraries included in an app that are collecting sensitive data,[8,13] often to the surprise of developers. After analyzing smartphone apps on PrivacyGrade.org, my research team found that about half of the apps that use location data only do so because a third-party library uses it, rather than the app itself.

From the app developer's perspective, it's difficult to understand the privacy-related behaviors of these third-party libraries. Beyond reading a lot of documentation, there's no easy way for a developer to understand what data a library uses and where it sends that data. Furthermore, there is no easy way to ensure that a distributed system overall has the desired properties. For example, an app might run on a smartphone, be managed by an operating system, use different hardware and software components, and send data to cloud services. How can privacy properties be guaranteed for parts or even all of this system?

This is an extremely difficult and open issue for cybersecurity in particular and software engineering in general. To some extent, Android's approach to having manifest files is one step forward. App developers have to state up front what sensitive data their app might access, in the form of permissions, such as "camera" and "call log." Can an approach like this be applied to pervasive computing? Are there ways of checking or even proving that an operating system or cloud service is enforcing these permissions correctly as information flows outside of its original device? Can trusted computing bases be used to help?

A related issue is that app developers and third parties need a sustainable way of generating revenue. Advertising-based business models are a popular approach, but they pose unique concerns for privacy because such services have a strong incentive to collect as much data as possible so as to serve up more personalized ads. While there are certain kinds of uses that are clearly unacceptable (such as showing job opportunities based on race), it is still unclear what sensitivities people have to different kinds of data and inferences under different situations. (I discuss this issue more in the section on users.) Also, are there better ways of sharing personal data with ad networks and other services so that they can deliver personalized content while also minimizing the sensitivity of personal data that is shared? One good example is Privad,[14] which pre-fetches ads so that users can see the ads, and the ad services can still collect click-through data, but they can't collect detailed behavioral information about users.

App Stores

Given the success of app stores for smartphones, it is very likely we will see similar stores for other pervasive computing platforms. These app stores also present a unique point of leverage for privacy as well, because they might facilitate privacy at scale.

For example, one research opportunity is to develop new techniques for examining the privacy-related behaviors of apps. A number of research projects have looked at static analysis,[15] dynamic analysis,[16] and even crowd analysis. With respect to crowd analysis, past work has used paid crowd workers to inspect screenshots and descriptions of apps[10,17] or get feedback from actual users of apps.[13] One drawback of dynamic and crowd analysis techniques is scalability, because they require more time, money, and human attention than static analysis approaches. As such, one research direction is to develop new ways of combining these approaches. A likely approach is to use static analysis on all apps, and then use more costly dynamic and crowd analysis for the most popular apps or selected apps flagged by static analysis.

Another research opportunity for app stores is to develop better ways of conveying privacy information to consumers. Today, the state of the art is to offer a long privacy policy full of legalese that few people (aside from lawyers) actually read. Computer-readable formats (such as P3P for

websites[18]) and visualizations (such as privacy nutrition labels[19]) are possible alternatives, but they have seen little adoption in practice. A challenge is that app stores want to balance between protecting users while also getting people to download more apps, and offering more privacy information might lead to fewer downloads. As such, this kind of privacy information might be better for third parties to offer, though there would also be many challenges here in offering up-to-date information.

OS and Middleware Manufacturers

With respect to operating systems, there has been a great deal of work looking at privacy issues for the OS of individual devices, particularly smartphones.[16,20] Here, I instead focus on middleware systems for managing multiple devices for homes or entire buildings.

A core issue here is helping people manage privacy at scale. As mentioned earlier, many low-end devices will have few if any software updates. While this issue is more related to cybersecurity than privacy, poor security can lead to accidental leaks of sensitive information. What kinds of mechanisms can we build so that our middleware can help keep software on low-end devices up to date? Complementarily, can we build formal models or infer models of normal versus abnormal behavior?

One possibility is to consider how to do a division of labor between the middleware and individual devices. For example, one option is to have low-end devices specify a great deal of metadata that the middleware can use as hints. One simple example is to have a URL that points to software upgrades, making it easy to check all devices to see if they have the latest versions. Contrast that with today's process of searching for the manufacturer's webpage, searching for the product ID, downloading the software onto one's PC, and then installing the updates onto a device. Another simple example is to have a description of what kinds of sensors the device has, what network services it connects to, and for what purposes. An early example of models for device behaviors are Manufacturer Usage Descriptions,[21] a draft IETF specification for letting manufacturers define normal behaviors. This kind of metadata can help in generating appropriate notifications and can help the middleware understand if the device is operating within normal parameters. What other kinds of metadata might be useful to help middleware with managing privacy?

Another core issue for middleware is helping developers. For example, can we offer APIs that make it easier for developers to balance between privacy and utility? As one example, in many situations, it is likely that people will be more willing to share that they are at "home" or "work" (just the string label) versus their exact GPS location. As another example, it is likely most people will be more comfortable with a pervasive computing app asking how loud it is in a given room rather than accessing the raw audio stream. These kinds of programming abstractions offer developers a clear benefit in that the developers don't have to know anything about machine learning but can still get the main information they want. These abstractions might synthesize or summarize data by space, time, or granularity, and they might help incorporate some model of user preferences to help determine sensitivity of the queried data.

Government and Third Parties

One underexplored issue for privacy in the context of pervasive computing is in empowering government agencies and interested third parties, such as journalists or advocacy groups, to help take the lead in pinpointing privacy problems and offering meaningful alternatives. With respect to government, one possible area of research is privacy for children. Many countries have laws that govern what kinds of data can be collected about children. Unlike other aspects of privacy, privacy for children is clear cut and thus might offer guidance in advancing privacy in general. One research issue here is developing scalable methods for identifying apps for kids and determining if those apps have inappropriate tracking behaviors.[22] Can we develop similar kinds of analysis tools for devices, particularly for toys, to understand their behaviors and ensure they're complying with existing laws?

More generally, there are two kinds of research that could help governments in either regulating or fining privacy violations. The first is in developing better kinds of analysis tools that can help government agencies understand the specifics of an app's or device's privacy-related behaviors. Much of this kind of analysis today is done manually and often by lawyers rather than computer scientists, so tools that can help novices greatly advance things. The second is in measuring uses of personal data to ensure that devices and network services are behaving appropriately. An example might be continuously evaluating advertising networks to understand what data they collect from pervasive computing devices and how they use that data—for example, ensuring they're complying with their stated policies and not discriminating based on race. Given that pervasive computing devices will likely all be running different software, the only scalable approach is to monitor network traffic, which leads to a host of research questions, such as how to identify specific devices if there are many devices, how to spot potentially sensitive data in network traffic, and how to reliably determine cause and effect, all in a scalable manner. As a concrete example, is there a general tool that could help analysts quickly understand that Samsung's Smart TV[23] monitors voice data, or what exactly Mattel's Talking Barbie[24] does when you talk to it?

Interestingly, the same kinds of research mentioned—better analysis tools and better measurements—are just as useful for third parties interested in privacy. In particular, journalists are an interesting case, because they are

often interested in exposing privacy surprises, and they offer a unique angle for potentially improving privacy, in terms of being able to shame egregious services or devices.

Users

Although there have been many user studies on privacy in the context of pervasive computing, there is still much that we don't know. For example, in 2012, *The New York Times* published an article that described how the store Target guessed if a young teenage girl was pregnant, using this data to deliver printed ads offering discounts for baby-related products.[25] This article led to negative comments from people worried about privacy and the growing amount of data that companies have about us. However, from a research perspective, we currently have little insight as to why people felt so concerned. Was it the fact that Target collected this data, or that they used the data to infer pregnancy? Or was it that they sent coupons for pregnancy-related products, or that a teenager's father found out she was pregnant through these coupons? Currently, all of these issues fall under the broad umbrella of privacy, and unpacking them could help with appropriate designs, interventions, or policy decisions to address people's concerns.

Similarly, for pervasive computing systems, we need better qualitative and quantitative data to understand the exact nature of people's concerns, so that we can legitimately address those concerns and draw clear lines about what is and is not acceptable. A deeper understanding of these issues would greatly improve our ability to design pervasive computing systems. For example, how concerned are people in general about different kinds of data types, such as location data, video streams, or sleep data, as well as combinations of data? Following up, how does a person's "privacy calculus" change based on the different perceived purposes of data use? For example, in past studies, Jialiu Lin and her colleagues found that people were very concerned about contact list data being used for advertising, but mostly neutral about location data being used for social media.[10]

If we can develop such a privacy calculus, can we also use it to develop useful defaults? Defaults are an important but often overlooked issue for privacy. For example, if all sensed data is private by default, the pervasive computing deployment will likely fail because of underutilization, low perceived value, and high burden in terms of users having to constantly allow apps to access personal data. In contrast, if all sensed data is publicly visible, there will likely be protests and potentially even outright rejection of systems due to legitimate privacy concerns. If we can develop good defaults for what data should be shared with whom and when, it can likely reduce privacy concerns and reduce user burden in terms of configuration, while offering app developers and users of the deployed pervasive computing systems a basic level of utility.

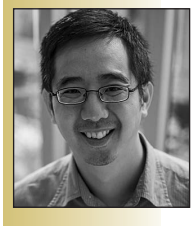
Does gender play a role in people's perceptions of privacy? For example, are women more likely to share data if it benefits groups or society in general? Also, are women more likely to be worried about tracking, in part due to concerns about stalking? Similarly, do people from different cultures have different conceptions and concerns about privacy? Thus far, the vast majority of research systems and studies have been conducted on WEIRD people[26]—that is, Western, Educated, Industrialized, Rich, and Democratic. Few studies have examined these kinds of cultural issues for privacy.

How do people's concerns about privacy change over time? When landline telephone systems were first deployed, many people objected to having landline phones in their homes,[27] because it "permitted intrusion... by solicitors, purveyors of inferior music, eavesdropping operators, and even wire-transmitted germs." Are there better ways to predict how people's attitudes and behaviors might change?

Lastly, policy makers have also been increasingly adopting the notion of contextual integrity[28] as a working definition of privacy, which looks at appropriate flows of information for specific contexts of use based on norms and values. For example, it makes sense to share personal health data with doctors but not necessarily with coworkers. However, one major challenge is in aligning often-rigid pervasive computing systems with fluid notions of context and norms. Are there scalable ways of using sensor and log data to help operationalize contextual integrity?

Here, I have highlighted some major privacy challenges and research opportunities, but it is still an open question as to how best to achieve privacy in practice. Furthermore, I would be remiss if I didn't point out that there are many other privacy challenges in specific domains of pervasive computing, such as lifelogging, drones, and social media.

the AUTHOR



Jason Hong is an associate professor at Carnegie Mellon University. His research interests include usable privacy and security as well as mobile and ubiquitous computing. Hong received his PhD from the University of California at Berkeley. He is a member of ACM and IEEE. Contact him at jasonh@cs.cmu.edu.

We are currently at a crossroads, not just in computing but in history. There is only one point in time when the foundation is laid for how computation, communication, sensing, and actuation will be woven into our physical world, and that time is now. These pervasive computing technologies offer tremendous opportunities in terms of healthcare, safety, sustainability, education, and more. But this vision is possible only if we can find ways of addressing the privacy issues and fostering trust by building systems that respect people as individuals, that offer people tangible value and provide the right level of control and feedback, and that do what people expect them to do. We are the ones who will be building these kinds of systems, so let's make sure we create a connected world that we all want to live in.

REFERENCES

1. M. Weiser, "The Computer for the 21st Century," *Scientific Am.*, Sept. 1991, pp. 66–75.
2. J.L. Boyles, A. Smith, and M. Madden, "Privacy and Data Management on Mobile Devices," Pew Research Center, 5 Sept. 2012; www.pewinternet.org/2012/09/05/privacy-and-data-management-on-mobile-devices.
3. M. Satyanarayanan, "Pervasive Computing: Vision and Challenges," *IEEE Personal Comm.*, vol. 8, no. 4, 2001, pp. 10–17.
4. T. Yu et al., "Handling a Trillion (Unfixable) Flaws on a Billion Devices: Rethinking Network Security for the Internet-of-Things," *Proc. 14th ACM Workshop on Hot Topics in Networks*, 2015, article no. 5.
5. M. Langheinrich, "A Privacy Awareness System for Ubiquitous Computing Environments," *Proc. 4th Int'l Conf. Ubiquitous Computing*, 2002, 237–245.
6. S. Lederer, J.C. Mankoff, and A.K. Dey, "Who Wants to Know What When? Privacy Preference Determinants in Ubiquitous Computing," *Proc. Extended Abstracts on Human Factors in Computing Systems* (CHI), 2003, pp. 724–725.
7. J. Wiese et al., "Are You Close with Me? Are You Nearby? Investigating Social Groups, Closeness, and Willingness to Share," *Proc. 13th Int'l Conf. Ubiquitous Computing* (Ubicomp), 2011, pp. 197–206.
8. R. Balebako et al., "The Privacy and Security Behaviors of Smartphone App Developers," *Proc. Workshop Usable Security* (USEC), 2014; http://repository.cmu.edu/cgi/viewcontent.cgi?article51278&context5hcii.
9. S. Consolvo et al., "Location Disclosure to Social Relations: Why, When, & What People Want to Share," *Proc. SIGCHI Conf. Human Factors in Computing Systems* (CHI), 2005, pp. 81–90.
10. J. Lin et al., "Expectation and Purpose: Understanding Users' Mental Models of Mobile App Privacy through Crowdsourcing," *Proc. 2012 ACM Conf. Ubiquitous Computing* (Ubicomp), 2012, pp. 501–510.
11. J. Yang, K. Yessenov, and A. Solar-Lezama, "A Language for Automatically Enforcing Privacy Policies," *ACM SIGPLAN Notices*, vol. 47, no. 1, 2012, pp. 85–96.
12. C. Dwork, "Differential Privacy: A Survey of Results," *Proc. Int'l Conf. Theory and Applications of Models of Computation*, 2008, pp. 1–19.
13. Y. Agarwal and M. Hall, "ProtectMyPrivacy: Detecting and Mitigating Privacy Leaks on iOS Devices Using Crowdsourcing," *Proc. 11th Int'l Conf. Mobile Systems, Applications and Services* (MobiSys), 2013, pp. 97–110.
14. S. Guha, B. Cheng, and P. Francis, "Privad: Practical Privacy in Online Advertising," *Proc. USENIX Conf. Networked Systems Design and Implementation*, 2011, pp. 169–182.
15. R. Pandita et al., "WHYPER: Towards Automating Risk Assessment of Mobile Applications," *Proc. 22nd USENIX Security Symp.*, 2013; www.enck.org/pubs/pandita-sec13.pdf.
16. W. Enck et al., "Taintdroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," *Proc. 9th USENIX Conf. Operating Systems Design and Implementation* (OSDI), 2010, pp. 1–6.
17. J. Lin et al., "Modeling Users' Mobile App Privacy Preferences: Restoring Usability in a Sea of Permission Settings," *Proc. Symp. Usable Privacy and Security* (SOUPS), 2014; http://repository.cmu.edu/cgi/viewcontent.cgi?article51275&context5hcii.
18. L.F. Cranor et al., *The Platform for Privacy Preferences 1.0 (P3P1.0)*, W3C specification, Apr. 2002; www.w3.org/TR/P3P.
19. P.G. Kelley et al., "A 'Nutrition Label' for Privacy," *Proc. 5th Symp. Usable Privacy and Security* (SOUPS), 2009, article no. 4.
20. Y. Jing et al., "RiskMon: Continuous and Automated Risk Assessment of Mobile Applications," *Proc. 4th ACM Conf. Data and Application Security and Privacy* (CODASPY), 2014, pp. 99–110.
21. E. Lear, R. Droms, and D. Romascanu, "Manufacturer Usage Description," IETF Internet draft, work in progress, Sept. 2016.
22. M. Liu et al., "Identifying and Analyzing the Privacy of Apps for Kids," *Proc. 17th Int'l Workshop Mobile Computing Systems and Applications*, 2016, pp. 105–110.
23. C. Matyszczyk, "Samsung's Warning: Our Smart TVs Record Your Living Room Chatter," *CNet News*, 8 Feb. 2015; www.cnet.com/news/samsungs-warning-our-smart-tvs-record-your-living-room-chatter.
24. S. Kim, "Here's What It's Like Playing with the Talking 'Hello Barbie,'" ABC News, 17 Sept. 2015; http://abcnews.go.com/Business/playing-talking-barbie/story?id533806499.
25. C. Duhigg, "How Companies Learn Your Secrets," *New York Times Magazine*, 2012; www.nytimes.com/2012/02/19/magazine/shopping-habits.html.
26. J. Henrich, S.J. Heine, and A. Norenzayan, "The Weirdest People in the World?" *Behavioral and Brain Sciences*, vol. 33, no. 2/3, 2010, pp. 1–75.
27. C. Fischer, *America Calling*, Univ. of California Press, 1994.
28. H. Nissenbaum, "Privacy as Contextual Integrity," *Washington Law Rev.*, vol. 79, no. 1, 2004, pp. 119–158.