Sensor Tricorder: What does that sensor know about me?

Gabriel Maganis*, Jaeyeon Jung†, Tadayoshi Kohno‡, Anmol Sheth† and David Wetherall‡†
University of California, Davis* Intel Labs Seattle† University of Washington‡
gymaganis@ucdavis.edu, {jaeyeon.jung,anmol.n.sheth}@intel.com, {yoshi,djw}@cs.washington.edu

ABSTRACT

As rich sensing applications become pervasive, people increasingly find themselves with limited ability to determine what sensor data the applications are collecting about them and how the applications are using the sensor data. Openness and transparency serve as our guiding principles in designing the Sensor Tricorder, a system that enables people to query third party sensors with their smartphones in order to learn about the data collection activities and privacy policies of the applications using the sensors. We leverage the increasing ubiquity of QR Codes in mobile applications and utilize them in a novel way. Our prototype system uses *active* QR Codes to visually communicate dynamic data such as the sensor activities and application privacy policies to smartphone users. Based on our experiences in building this prototype, we identify the key properties that sensor platforms must provide to support transparency and openness and highlight the main challenges involved in realizing these properties.

1. INTRODUCTION

Rich sensing devices equipped with cameras and microphones are increasingly adopted by new services and products used in everyday environments such as homes, offices and public spaces. The Microsoft Xbox, for example, uses the Kinect [4] add-on (a sensor device capable of full-body 3D motion capture, facial recognition and voice recognition) as the basis for easy-to-use, interactive games and services. Sensor-based domestic elder care systems leverage distributed motion detectors, cameras and other sensors situated in the living space to infer daily activities. Video conferencing systems make use of distributed cameras and microphones in the room to provide interactive communication with remote parties. We can only expect these kinds of sensing-based products and services to become widespread as they provide new functionality that consumers want.

However, such sensor-rich pervasive computing environments put people's privacy at risk. Sensors collecting information about people in its range could have undesirable outcomes such as disclosure of sensitive conversations and leaks of private moments. Despite this, people are provided with limited or no visibility into what data about them are recorded and how the data are used. Consider the example of a smart conference room outfitted with cameras, microphones and speakerphones used for recording meetings and video conferencing. With current technology, when a person walks into a room, the only readily available information they can glean from the sensors is whether sensors are currently on or off (i.e., through the LED indicator lights commonly found on these devices). However, beyond this limited information, sensors are black boxes that provide no additional information about how long they have been recording, what data is being collected or how the data is being used. Such continued lack of transparency creates a barrier to the adoption of sensing environments where sensors may be used by a wide range of applications.

A common approach is to use an LED on the sensor to indicate whether the sensor is recording or not and this fails to provide the transparency needed to address the above problem. This convention suffices when the sensor functionality is simple and well understood (e.g., take a picture and store to local storage). However, as sensors become connected, have multiple applications using them and can potentially be controlled remotely, such an approach becomes insufficient.

In this paper, we argue that the Principle of Openness [17] – which states that policies and procedures regarding how sensor data is used should be readily available to users – is the key consideration with respect to building transparent sensing systems. We outline three properties that sensors should seek to achieve. First, sensors should be able to maintain a log of their activity. Second, sensors should have access to the privacy policies of the applications using them. Finally, the third property is that sensors should make their activity log and the application privacy policies easily accessible to the people affected by them. These properties taken together enable users to be more aware of how sensing environments are making use of their personal data.

To better understand the challenges involved in realizing the requirements for supporting transparency, we have built an initial prototype that seeks to achieve them on a sensor platform[1]. Our prototype system, called *Sensor Tricorder*[2], enables users to query sensor platforms with their smartphone to receive a privacy report containing a record of applications' data collection activities along with their privacy policies regarding the collected sensor data. A key feature of the sensor tricorder is that it makes the privacy report easily accessible to smartphone users through the visual com-

HotMobile 2011 March 1-2, 2011, Phoenix, Arizona USA
Copyright 2011 ACM 978-1-4503-0649-2 ...$10.00.

[1] In the rest of this paper we use the terms sensor or sensor platforms interchangeably to refer to the sensor system that supports the transparent sensing requirements.

[2] "In the fictional Star Trek universe, a tricorder is a multi-function handheld device used for sensor scanning, data analysis, and recording data." – Wikipedia

munication channel. The privacy report is periodically updated, encoded into a QR Code and then displayed on the sensor itself. We refer to these dynamically changing QR Codes as *active* QR Codes. Then, users simply need to point their camera phone at the sensor to decode the latest information using any standard QR Code reader application.

There are several alternatives to using QR Codes. However, unlike wireless communication technologies such as Bluetooth or Wi-Fi that require dedicated out-of-band mechanisms for device pairing and controlling range, the use of the visual channel significantly simplifies pairing and the mapping of sensors to physical locations. The visual channel is a suitable proxy to the physical scope of sensors like cameras and microphones that are confined by walls and doors and is well aligned with users' mental models of privacy provided by these physical barriers. To our knowledge, the sensor tricorder system is the first to propose the use of active QR Codes to visually communicate the activity logs of sensor platforms as well as the privacy policies of the applications using them. QR Codes provide an efficient and practical means to communicate the most data to users through the visual channel.

Based on our experience of building the sensor tricorder prototype, we encountered several open research challenges. First, maintaining a trusted sensor activity log without requiring a large trusted base is challenging. Second, simultaneously enabling openness and access control is not straightforward, especially when there are multiple stakeholders involved in environments like homes and offices. Lastly, it is challenging to enable transparency while preserving privacy at the same time. We highlight these research challenges that we hope to address as future work.

The rest of the paper is structured as follows. We begin by presenting the three main properties required to realize the sensor tricorder in Section 2. We make our threat model explicit in Section 3. We then describe the design and implementation of the sensor tricorder prototype in Section 4 and highlight the research challenges in Section 5. Finally, we present related work in Section 6 and conclude in Section 7.

2. REQUIREMENTS FOR TRANSPARENCY

The Principle of Openness (or, Notice) [17] provides a simple, intuitive guideline in designing privacy-aware ubiquitous systems. It states that policies and procedures regarding how sensor data is used should be made readily available to people who are affected by the sensor system. This principle is an interpretation of the Notice/Awareness code as part of the Fair Information Practice principles [1] and is applied in many situations today. For example, most legal systems require establishments that employ surveillance cameras to properly notify their customers that they are being recorded. Sensing devices such as cameras and microphones themselves employ some form of notification mechanism. Most digital cameras have LED status indicators that light up (usually green or red) and make prominent clicking sounds to indicate activity such as video recording or taking a picture. While these simple notification mechanisms are good examples, they would not be able to provide the transparency needed for increasingly rich and connected sensing environments.

Consider an intelligent conference room equipped with cameras, microphones and display screens that can be used by multiple applications (e.g., video conferencing application, an application that records talks, automatic light on/off controller based on people's presence, etc). A person entering such a room may have several questions about the sensors that are difficult to answer using existing mechanisms:

1. Am I being recorded now? If so, which sensors are being used by which applications? How long have I been recorded since I entered the room?

2. Is the recording stored locally or transmitted somewhere by the applications? If so, what are the applications' data retention policies?

3. What was the configuration of the sensors when they were on? (e.g., was the microphone able to capture my whispering to co-workers?)

Similar questions can be asked by a person at home, sitting in the living room where a video game console is set up with cameras and microphones. From these user-centered questions grounded in real situations, we identify three main requirements for realizing the principle of openness in sensor platforms.

Sensors should log their activity. First, there needs to be a log of sensor activity. Although the camera and microphone would likely have LED lights to indicate whether they are recording or not, they would clearly not be sufficient to answer questions such as (1) and (3). Indeed, even if people were only interested in knowing whether the sensors are simply on or not, current status indicators could be easily missed. The LED would light up when recording happens for a few seconds but turn off before people can realize that the sensor was recording them. This problem necessitates a recorded history of sensor activity as opposed to only their current status.

Sensors should maintain the privacy policies of the applications using them. A log of sensor activities would be limited to identifying *when* the data was collected and *who* collected the data. Having access to the privacy policies of the applications using the sensors would answer question (2); including detail about the data collection activity such as *what* data was collected, for *whom* it is being collected, and *why* it was collected.

The activity logs and privacy policies should be readily accessible to the user. Although there will always be sensors that are hidden for malicious purposes, the sensor platform must make sure that its users are informed about what is going on around them. The platform could either broadcast the information or require users to query for it explicitly. Whether it is one or the other, the information should be available in an easily accessible format.

The Sensor Tricorder embodies the above three properties in a novel way as follows: (1) a log of sensor activity is maintained, (2) then the privacy policies of the applications using the sensors are used to augment the information provided by the log and produce a more meaningful report about the sensing environment, (3) and finally the report is *visually* communicated to the people affected by the sensor platform through dynamically changing QR Codes.

3. THREAT MODEL

Our primary goal is to improve the state of openness and transparency for *compliant* sensing systems. Rogue or malicious sensors are a very challenging problem since they can always be hidden in the environment, lie about their state or provide bogus data. Hence, we assume that the sensor hardware and the software that controls the sensor are trusted. On the other hand, we assume that the applications using the sensors are untrusted. This assumption is similar to that of Apple's iPhone and the Android mobile platform in which downloaded, third party applications run on trusted firmware.

The main threats we are concerned about are:

Unauthorized modification. We want to prevent unauthorized modification of the sensor activity logs and preserve its integrity.

Spoofing. We want to prevent users from receiving fake or illicitly modified privacy reports.

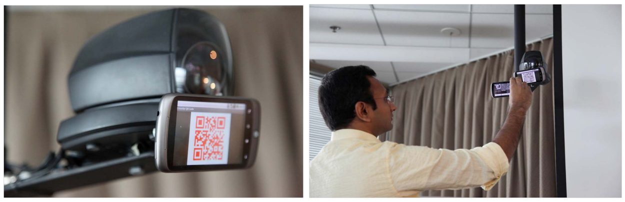

Figure 1: The QR Code encodes the camera's recent activity and data retention policy (left). An individual reviews the privacy report using a barcode reader mobile application (right).

Unauthorized access. We want to limit access to the privacy reports to users who are affected by the sensors i.e., within proximity.

We do not directly address private data leaks although they are an important consideration as we discuss in Section 5.

4. THE SENSOR TRICORDER

Figure 1 shows our prototype of the sensor tricorder system. The activity logs and privacy policies of the applications using the sensor are made available to the user by displaying the information as a QR Code attached on the sensor itself. Users query these sensors by pointing their smartphone at the QR Code and decoding it with any standard QR Code scanning application. In a setting where sensors are too small or are being controlled by a central server in the room, the QR Code can reflect an aggregate of the information from the sensors and be displayed on the central server itself.

In the rest of this section we discuss the design requirements of the three main components of the tricorder system and present an overview of the implementation of the sensor tricorder prototype.

4.1 Sensor Activity Logs

A foundational piece of the sensor tricorder system is a system-wide logging component running on the sensor platform. It is responsible for maintaining a complete, accurate, and ordered log of all usage of the sensors by applications. Additionally, the logging component should record all configuration changes made to the sensors by the applications. Thus, the logging component serves as a *trusted* shim layer (independent of the applications) that is able to intercept all application specific events that change the state of the sensor.

There are two main requirements involved in designing such a logging component. First, the logging component must be tamper-proof and other software components or applications using the sensor platform should not be able to modify the log. Otherwise, misbehaving applications could make changes to the log and remove entries that reveal their malicious behavior. Hence, the tricorder system should employ necessary mechanisms to ensure that the log's integrity (and potentially confidentiality) is intact.

The second requirement is for the logging component to have a complete view of sensing events occurring on the sensor platform. This involves determining at what layer of the software stack the logging component should reside. It could be anywhere from the application level, the operating system level, the device driver level or even at the firmware level. In general, there is more context information (e.g., application name, remote server name) available when implementing the logging component at higher levels of the stack.

4.2 Sensor Specific Privacy Policies

Sensor activity logs are able to identify *what* sensor data is being collected by an application but would not be able to determine *how* the application is using the sensor data. To address this, the applications using the sensors must be required to provide their privacy policies such as those provided by websites. The tricorder system uses the privacy policies of the applications to enhance the information available from the sensor activity logs. Likewise, the sensor activity log gives context to the privacy policies of the applications.

We leverage existing work on a machine-readable format of privacy policies developed by the Platform for Privacy Preferences Project (P3P) [3]. However, P3P policies are verbose and make it cumbersome for the user to effectively find relevant information. Hence, the primary design requirement for using P3P-like machine-readable policies is to provide a concise representation of the privacy policy that is relevant to the sensor being queried.

4.3 Sensor Privacy Reports

An essential requirement for the sensor tricorder is to effectively communicate the sensor activities and privacy policies of applications ("privacy reports" hereafter) to users. There are many potential approaches such as RF-based wireless communication, infrared light based data transfer, or even visual light based communication such as QR Codes or blinking LED lights. We systematically review each of these techniques according to our design goals and summarize the results in Table 1. We note that these values report the state-of-the-art for each of the methods listed.

The *data rate* determines how long the user must engage in communication with the tricorder hence, is an important consideration for the tricorder system. To limit access to privacy reports to users within proximity of the sensors, the *range* and device *pairing* attributes must be considered. Service *discovery* pertains to how easy it would be for users to find out about the privacy reports upon entering an environment where a tricorder is deployed. The *visibility* attribute is an important consideration as a human factor because it helps improve people's awareness of sensors in the environment. For example, although in a different setting, a study showed improvement in people's awareness of privacy risks associated with the use of unencrypted WiFi networks when a stock ticker like display (in real time) showed personal information broadcasted over the network [9].

Radio Communication: Radio based communication techniques like Bluetooth and WiFi are now pervasive across a wide range of consumer electronics and provide high data rates. However, due to the broadcast nature of the wireless medium it is difficult to map the radio signals to a physical region. This makes pairing and range difficult to control. Additionally, these techniques often require out-of-band mechanisms to address these problems. Near Field Communication (NFC) is another RF-based communication technique but requires the sensors to be in very close proximity ($<$ 0.2 meters) of the user.

Infrared Communication: Infrared based data transfer is both directional, reasonably ranged and would require a line-of-sight path between the sensor and user. However, the inability to visualize the communication hinders discovery and makes pairing difficult in situations where multiple sensors are deployed right next to each other.

Visible Light Communication: Visible light based communication techniques such as blinking LEDs and QR Codes [6] are desirable because they are aligned with users' mental models of privacy provided by physical barriers such as doors, walls and windows, and enables explicit pairing and discovery. Visible light based communication also provides proximity based access control as the user needs to be in line-of-sight of the sensor within some bounding distance. These methods do have their own limitations however. For example, cameras with a zoom feature could easily violate the

	QR Codes	Blinking LED	WiFi	Bluetooth	Infrared	Near Field
Data Rate	6 Mbps	500 Mbps	300 Mbps	24 Mbps	1024 Mbps	848 Kbps
Range	5m	5m	250m	100m	1m	<0.2m
Pairing	Easy	Easy	Hard	Hard	Easy	Easy
Discovery	Hard	Hard	Easy	Easy	Hard	Hard
Visible	Yes	Yes	No	No	No	No

Table 1: Comparing QR Codes with other potential approaches. The state-of-the-art for each technology is given.

proximity rule and we discuss this issue further in Section 5.2.

Why QR Codes? The blinking LED based approach is an emerging technology that makes use of a large number of LEDs that toggle at a high frequency that is not perceivable by the human eye. While this is a promising approach, decoding the information requires specialized cameras on the user's device that are not available on commodity computing platforms. On the other hand, QR Codes are becoming ubiquitous due to its popularity as an advertising medium and growing support on mobile platforms. Furthermore, compared to methods such as simply displaying the text of the privacy report, QR Codes are able to encode up to 4,296 alphanumeric characters in a single barcode. We note that this is characteristic to 2D barcodes in general however, QR Codes seem to be the most popular and accessible 2D barcode algorithm at the moment. Hence, we use QR Codes that are lightweight to encode and decode and can leverage existing commodity cameras and displays.

Pervasive applications have traditionally used QR Codes to simply encode static web links to which users are redirected to. We utilize them in a novel way and *dynamically* generate them to reflect the current state of the sensing environment. Moreover, using QR Codes allow the decoding application to be customized e.g., to display the encoded information in a more usable and friendly manner. Recent research has also shown that QR Codes can be made significantly smaller, can be read from distances as far as 10 to 15 meters [21], and can also support high data rates [22]. We refer interested readers to [14] for a more detailed discussion on 2D barcodes and other pervasive applications.

4.4 Prototype Implementation

This section presents an overview of the implementation of our prototype. Commodity sensor platforms are typically closed proprietary systems therefore, we implement the sensor tricorder using the Android 2.1 platform. Thus, each sensor in our system runs in the Android operating system and implements the three components of the tricorder described in the previous section.

4.4.1 Logging Sensor Activities

We integrate the logging component into the operating system (OS) controlling the sensor platform. This allows us to leverage traditional OS mechanisms for access control. The Android architecture serves us well since all access to sensors on the phone is mediated by the operating system. This allows us to carefully place logging statements in the privileged Android source code and detect all application level requests to use sensor data.

The sensor activity log is a fixed-sized circular append-only buffer implemented as a Linux kernel device driver. The circular buffer is exported as a `/dev/log/main` device and applications or other non-privileged code cannot directly access the buffer without proper permissions. In addition to sensor activity, the logging infrastructure also logs the process ID of the application that is accessing the sensor. This helps in managing different application requests to use the same sensor. The tricorder can then query the `/dev/log/main` device and identify the relevant sensing events.

4.4.2 Privacy Policy

In order to make P3P policies concise and relevant to the sensor being queried by the user we make two key changes to the way P3P policies are integrated with the tricorder system. First, the P3P specification structures policies in a hierarchical manner where the actual data being described is at the bottom of the hierarchy. The hierarchical structure does not lend itself to retrieving the policy given a piece of data. For the tricorder, it is more desirable to specify policies in a flatter structure such that given a type of data (e.g., photo), it could easily find the privacy policy pertaining to it. Hence, privacy policies in the tricorder system are specified simply as tuples of the data together with its properties.

The second observation is motivated by the "Nutrition Label" for privacy [15] concept. We include all of the required properties contained by the `<STATEMENT>` elements of P3P policies. More specifically, we choose to describe data by their purpose, recipient and retention policy. Additionally, we also include the category property since it is commonly used by P3P policies when specifying collected data.

We refer interested readers to the P3P specification [3] for a more complete treatment of these properties but provide some examples here. The *category* property provides a hint to users about the intended use of the data. Categories include "physical" for information that allows an individual to be contacted or located in the real world, or "online" for information that allows an individual to be contacted or located on the Internet. The *purpose* property tells users the purpose of the data collection or the uses of the data. For example, a purpose of "current" lets one know that the data was collected to complete or support the activity for which the data was provided, while "admin" tells the user that the data was collected for administration purposes. The *recipient* property designates one or more entities who will receive the collected data. For example, a value of "public" means the data would be shared with anyone. Lastly, the *retention* property describes how long the data might be kept by the recipients. That is, the data might be retained "indefinitely", retained as long as required legally ("legal-requirement") or not stored at all ("no-retention").

Privacy Reports. When an application is installed on the sensor, the tricorder loads the privacy policy provided by the application in a simple list format. Each policy entry consists of the name of the application, the sensor it uses, the data it collects, and the four tuple of the category, purpose, recipient, and retention properties for the data. The tricorder then uses this policy database to extend the information available from the activity log. For each sensing event recorded by the logging component, the tricorder finds the corresponding privacy policy using the application's name and data type. Finally, the privacy report is generated as a list of events comprised of the timestamp of when an event occurred, the name of Android application that generated the event, the sensor that was used, the kind of data collected as a result of the event, and the application's policy towards the data.

Figure 2: The QR Code encoding for the most recent events in the following example privacy report:

```
3:31:02 PM|Skype|Connected|Photo/Video
          |Sensor Data|Current|Ours|No Retention
3:31:12 PM|Skype|Video started recording|Video
          |Sensor Data|Current|Ours|No Retention
3:40:03 PM|Skype|Picture taken|Photo
          |Sensor Data|Current|Ours|No Retention
3:47:03 PM|Skype|Video stopped recording
          |Video|Sensor Data|Current|Ours|No Retention
3:47:20 PM|Skype|Disconnected|Photo/Video
          |Sensor Data|Current|Ours|No Retention
```

4.4.3 Active QR Codes

Our sensor tricorder system uses dynamically generated QR Codes to communicate the privacy reports to people. The tricorder system periodically (i.e., every 5 minutes) builds a privacy report from the sensor activity logs and application privacy policies then encodes it into a QR Code. Hence, the QR Code being displayed on the sensor is periodically refreshed to reflect the most recent state of the sensing environment. In order to encode the privacy reports, the tricorder makes use of a well-known barcode image processing library called `zxing` [5]. Users can then query the sensor to learn about its recent sensing activities simply by pointing their smartphones at the QR Code and then decoding it with any standard QR Code reader application.

Figure 2 shows an example of a privacy report with the QR Code encoding of the most recent events for a simple application that periodically uses the camera to take a picture. We wrote a hypothetical privacy policy for this application.

The privacy report shown in Figure 2 addresses the questions posed in Section 2. The timestamps in the log show that the camera is currently off and was on for about 15 minutes in the past. The log entries include the application name that made use of the camera data (i.e., “Skype”). The data retention policy of the application encoded in the report shows that the camera data was not stored locally or remotely (i.e., “No Retention”). The sharing policy of the application is also encoded in the report and shows that the camera data is not shared to any third parties (i.e., “Ours”). Moreover, “Current” indicates that the application developer is not using the data for any other purpose other than the Skype application.

Clearly, users would not be interested in reading such detailed logs. However, tools like the “Nutrition Label” for privacy [15] could summarize this information in a more friendly format.

5. RESEARCH CHALLENGES

Our experiences in prototyping the sensor tricorder and applying it to different real-world scenarios uncovered a wide spectrum of research challenges ranging from trust, authorization, and the tradeoff between privacy and transparency. This section outlines some of these key challenges along with possible solutions.

5.1 Reducing Trust in Sensors and Applications

Trust is a major challenge for computer systems and sensor platforms are no exception. Our prototype of the tricorder includes the complete software stack that controls the sensors (i.e., all of Android) as part of the trusted computing base (TCB). Smaller TCBs are ideal because they are easier to check for errors or bugs, which means there are less vulnerabilities in the TCB. A key challenge is to be able to reduce the size of the TCB while enabling the three requirements for transparent sensing.

The requirement for a tamper-proof sensor activity log also presents a challenge. While a tamper-proof log would be difficult to achieve, there is a significant amount of research into tamper-evident logging which include a scheme that utilizes cryptographic hash chains [23], a scheme utilizing a tree-based data for efficient verification [10], and a technique that requires trusted platform modules [8].

The software that generates QR Codes from the activity logs must also be trusted. Other applications and software components should not be able to modify the QR Code. To address this, we hope to leverage existing research on trusted execution as well as reducing the size of the TCB [24, 20, 19]. Another promising approach is to integrate the logging component into the sensor platform hardware itself. Events detected by the logging component on the hardware platform like hardware interrupts, sensing a busy data channel or configuration changes to the sensor could be used to generate the activity log. We are exploring efficient ways of adapting these techniques to a typically resource-constrained sensor platform in our future work.

Privacy policies can inform users about what an application might do with their data, however, they are mere claims or statements (albeit with legal implications) and it remains a challenge to ensure that the applications adhere to their privacy policies. There have been several recent approaches to solve this problem. Privacy Oracle provides a framework for analyzing applications and finding privacy leaks using an approach suited for software testing [13]. A more active approach is to employ dynamic taint analysis and track the flow of sensor data [11] and enforce the policies specified by the application. Leveraging these additional verification steps can further enhance the trustworthiness of the privacy reports provided by the sensor tricorder.

5.2 Authorization

Although QR Codes can already provide some level of access control through proximity, some situations require stronger guarantees. To support more complex scenarios, utilizing cryptographic methods is one possible approach. For example, recall our conference room example for an office scenario, visitors might use the conference room as well but should not necessarily be allowed to read the contents of the QR Code. In this case, the QR Code may reveal information about proprietary company applications using the sensor platform. This simple binary policy could be implemented by having the sensor platform encrypt its privacy reports before encoding them into a QR Code. Hence, only employees who have the decryption keys would be able to read the sensor platform's reports. Even in situations in which more than a two-level access control is required, cryptography can certainly provide the technical tools for enabling access restrictions. However, as we will discuss in the following section, the main challenge lies within determining the right balance between privacy and transparency.

5.3 Balancing Transparency and Privacy

If not carefully designed, enabling openness and transparency could easily result in private information leaks in the system. Sensor activity logs are typically verbose and contain unnecessary details that must be filtered out or otherwise result in privacy leaks. For instance, if the sensor platform recorded unique identifiers for people captured by a camera and inadvertently made them available through the log, then the sensor tricorder could potentially enable tracking an individual's location. The unique identifier could show up through the investigation of the privacy reports from sensor platforms at different locations such as the home, the office, a coffee shop or even a shopping mall. Another example is that revealing details in sensor activity logs that have been previously unexposed by the system could create new privacy issues for the users of the sensing platform. The users may not be aware that their use of the sensing platform (e.g., which applications they choose to run, when, and for how long) could be revealed to anyone by the sensor tricorder. Deployments of a tricorder system should carefully consider issues such as the above.

Exploring other potential privacy risks introduced by the sensor tricorder involves understanding peoples' mental models and how people might interact with the system. One direction is to evaluate the prototype in a field study via interviews of the participants and logs of how the system was actually used. Another direction is to design and implement mechanisms to limit the exposure of detailed logs. For example, as our prototype does, privacy reports could only include the most recent sensing events by default (e.g., only the past 5 minutes). Furthermore, the system could even provide an option for authorized users to delete certain portions of the recorded history of sensing events.

6. RELATED WORK

There has been much research in designing ubiquitous computing systems that take user privacy into consideration. Perhaps the closest to our proposal is the privacy awareness system (*pawS*) described in [18]. Their system also recognizes the importance of an announcement mechanism and encourages openness by using P3P policies. However, they focus more on allowing users to keep track of their personal data through *privacy proxies*. Moreover, they use P3P policies directly and use Bluetooth radio or IrDa to announce their system's data usage policies. Similarly, the Sentry@Home system leverages the Smart Home to act as a privacy proxy for users and serve as a central enforcement point for accesses to sensitive data [7]. Hong and Landay take a different approach and build the Confab toolkit to help developers design ubiquitous computing applications that take privacy into consideration [12].

There is also research on developing ways to communicate privacy policies more effectively to users. Kelley et al. develop a "nutrition label" for privacy [15]. Their goal was to learn from existing mechanisms such as nutrition labels and design a visual presentation of privacy policies that would help improve users' understanding of them. In a much larger study, they found that the standardization of privacy policy formats improve the ability of users to find information and have positive effects on reader enjoyment of privacy policies [16]. In an effort improve the auditability of privacy policies of websites, a suite of tools called the IdM Policy Audit System was developed and released for the public [2].

The use of 2D barcodes in pervasive computing has become increasingly popular. Kato and Tan discuss this trend in [14]. They also analyze six 2D barcode algorithms and compare them using *first read rate* as a metric. In terms of novel applications, McCune et al. utilized 2D barcodes to solve several problems in computer security including secure device pairing.

7. CONCLUSION

The continued lack of transparency provided by sensor platforms put peoples' privacy at risk as rich sensing applications become more complex and pervasive. People increasingly find themselves with limited ability to determine how applications use the data gathered by the sensors around them. We advocate that in order to provide transparency, sensors should maintain a log of their activity, require privacy policies from the applications using them, and make this information readily accessible to the interested user. To enable this vision, we have developed a prototype of the Sensor Tricorder system which realizes these properties. Users can use their smartphones to decode the dynamically updated QR Codes on the sensors which reflect the sensor usage activities of applications as well as their privacy policies. Our experiences in building this prototype uncovered several unique challenges related to reducing the TCB for sensor activity logging, enforcing application provided privacy policies of sensor data use, building trusted displays for QR Codes, and maintaining a balance between transparency and privacy. We hope to address these challenges as part of our future work.

8. REFERENCES

[1] Fair Information Practice Principles. http://www.ftc.gov/reports/privacy3/fairinfo.shtm.

[2] IdM Policy Audit System. http://www.isoc.org/projects/idm_policy_audit_system.

[3] P3P 1.1 Specification. http://www.w3.org/TR/P3P11/.

[4] Xbox Kinect. http://www.xbox.com/en-US/kinect.

[5] Zxing 1D/2D Barcode Image Processing Library. http://code.google.com/p/zxing.

[6] ISO/IEC 18004: QR Code 2005 Bar Code Symbology Specification. Technical report, International Organization for Standardization, 2006.

[7] S. A. Bagues, A. Zeidler, F. Valdivielso, and I. R. Matias. Sentry@Home - Leveraging the Smart Home for Privacy in Pervasive Computing. *International Journal of Smart Home*, 1, 2007.

[8] B.-G. Chun, P. Maniatis, S. Shenker, and J. Kubiatowicz. Attested Append-Only Memory: Making Adversaries Stick to Their Word. In *SOSP*, 2007.

[9] S. Consolvo, J. Jung, B. Greenstein, P. Powledge, G. Maganis, and D. Avrahami. The Wi-Fi Privacy Ticker: Improving Awareness & Control of Personal Information Exposure on Wi-Fi. In *Ubicomp*, 2010.

[10] S. A. Crosby and D. S. Wallach. Efficient Data Structures for Tamper-Evident Logging. In *USENIX Security*, 2009.

[11] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones. In *OSDI*, 2010.

[12] J. Hong and J. Landay. An Architecture for Privacy-Sensitive Ubiquitous Computing. In *MobiSys*, 2004.

[13] J. Jung, A. Sheth, B. Greenstein, D. Wetherall, G. Maganis, and T. Kohno. Privacy Oracle: a System for Finding Application Leaks with Black Box Differential Testing. In *CCS*, 2008.

[14] H. Kato and K. T. Tan. Pervasive 2D Barcodes for Camera Phone Applications. *IEEE Pervasive Computing*.

[15] P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder. A "Nutrition Label" for Privacy. In *SOUPS*, 2009.

[16] P. G. Kelley, L. Cesca, J. Bresee, and L. F. Cranor. Standardizing Privacy Notices: an Online Study of the Nutrition Label Approach. In *CHI*, 2010.

[17] M. Langheinrich. Privacy By Design – Principles of Privacy-Aware Ubiquitous Systems. In *Ubicomp*, 2001.

[18] M. Langheinrich. A Privacy Awareness System for Ubiquitous Computing Environments. In *Ubicomp*, 2002.

[19] J. M. McCune, Y. Li, N. Qu, Z. Zhou, A. Datta, V. Gligor, and A. Perrig. TrustVisor: Efficient TCB Reduction and Attestation. In *IEEE Symposium on Security and Privacy*, 2010.

[20] J. M. McCune, B. Parno, A. Perrig, M. K. Reiter, and H. Isozaki. Flicker: An Execution Infrastructure for TCB Minimization. In *EuroSys*, 2008.

[21] A. Mohan, G. Woo, S. Hiura, Q. Smithwick, and R. Raskar. Bokode: Imperceptible Visual Tags for Camera-based Interaction from a Distance. In *ACM SIGGRAPH*, 2009.

[22] S. D. Perli, N. Ahmed, and D. Katabi. PixNet: LCD-camera Pairs as Communication Links. In *ACM SIGCOMM*, 2010.

[23] B. Schneier and J. Kelsey. Secure Audit Logs to Support Computer Forensics. In *ACM TISSEC*, 1999.

[24] E. Shi, A. Perrig, and L. V. Doorn. BIND: A Fine-Grained Attestation Service for Secure Distributed Systems. In *IEEE Symposium on Security and Privacy*, 2005.