NotiSense: An Urban Sensing Notification System To Improve Bystander Privacy

Sarah Pidcock, Rob Smits, Urs Hengartner and Ian Goldberg
Cheriton School of Computer Science
University of Waterloo
{snpidcoc,rdfsmits,uhengart,iang}@cs.uwaterloo.ca

ABSTRACT

The growth in popularity of hand-held mobile devices has fuelled research exploring how to harness the collective abilities of sensors attached to these devices. One area of development has been *urban sensing*, which explores building a crowd-sourced wireless sensor network using consumer mobile devices. Urban sensing participants use their devices to capture information about their surroundings to contribute to an urban sensing system. Existing research has explored protecting the privacy of the urban sensing participants, through anonymization and aggregation of collected data. We are interested in the privacy of *bystanders* who may be inadvertently affected by nearby urban sensing data collection. There are difficult aspects to this problem, as we must weigh the privacy of bystanders against the privacy of urban sensing participants. We describe *NotiSense*, a simple system that provides useful notifications of nearby sensing activities to those who choose to subscribe. We evaluate a prototype implementation of NotiSense and its use of Wi-Fi to provide notifications. NotiSense is a good approach to enhancing the privacy of bystanders and opens up interesting challenges for future work.

1. INTRODUCTION

Alice has lost her cat in an unfamiliar neighbourhood. She creates a task in an *urban sensing* service that asks volunteers to submit photos and locations of local wandering cats. The urban sensing paradigm is an established domain of mobile wireless sensor networking research and has gained in popularity over the last several years [8]. In architectures based on this paradigm, groups of mobile devices are leveraged to collect contextual data from the surrounding environment that otherwise would be difficult to obtain.

An urban sensing network is built through users of mobile devices registering as participants with an urban sensing service. Urban sensing services are either designed for a certain type of sensing task or may support arbitrary types of tasks. Existing research also distinguishes between participatory and opportunistic urban sensing [8]. The former requires the carrier of a mobile device to explicitly choose which tasks to complete, whereas the latter has devices sense without user interaction on a potentially continuous basis depending on the participant's preferences. An urban sensing service accepts sensing tasks from an urban sensing application and disseminates these tasks to the mobile devices. Human participants carry these devices, which collect data using built-in or attached sensors and report their findings as directed by an urban sensing task. Data analysis and aggregation may be performed on submitted reports before they become accessible by the urban sensing application that submitted the task.

The development of urban sensing has faced challenges relating to privacy. Existing research (e.g., [2, 5, 6, 7, 12, 15]) focuses on the privacy of the human carriers of mobile nodes—the urban sensing *participants*. This research explores privacy protection through anonymization and aggregation of collected data. Protecting the privacy of the data collectors is of vital importance because sensitive data is contained in location, audio recordings, photographs, videos, Bluetooth signals, and a multitude of other data forms that are supported by urban sensing. However, the privacy of *passive bystanders* also needs to be considered, as sensitive information relating to them could be recorded by an urban sensing application that is active in the area. Consider the earlier example about Alice's lost cat. Volunteers who are taking pictures of wandering cats may also inadvertently capture images of bystanders. The privacy of these bystanders has so far been overlooked in urban sensing research.

The privacy of bystanders has the potential to be impacted the most in implementations that involve constant recording of data from the surrounding environment. One example of an urban sensing application that could cause such privacy problems is *BikeNet* [4], which employs a large collection of sensors to measure not only cyclists' personal states (e.g., heart rate, wheel speed) but their cycling experiences (e.g., noise level, pictures of a route) and upload the sensed data to a repository. *Biketastic* [13] follows a similar approach, but uses only the sensors available in a mobile phone for sensing. In these applications, the sensed and uploaded data, such as video and audio data, may include information about bystanders and violate their privacy.

Existing urban sensing architectures do not suggest any standards for making the general public aware of current or future sensing activities. The quiet collection of large amounts of contextual data is likely to cause distrust for urban sensing among the general public. Without notification, urban sensing is not much different from recording people without their knowledge, a concept with which people are uncomfortable. Adding transparency to urban sensing architectures will give people the power to exclude themselves from urban sensing datasets by avoiding areas where sensing activities are taking place or to voice their concerns about unacceptable activities or unreasonable amounts of activity. Social acceptability of urban sensing must be improved before it can become widely accepted.

One approach to protect the privacy of bystanders is to anonymize collected data. For example, Google Street View, which may also violate the privacy of bystanders, currently blurs the faces of passive bystanders. However, this approach, with 98% effectiveness [11], is not perfect. Moreover, partial face detection, in the case where only part of a face is captured, is generally less reliable. The potential failure of anonymization has motivated us to look at a notification-based solution for protecting the privacy of passive bystanders. Note that in practice we expect that both anonymization-based and notification-based solutions will be deployed, maybe in parallel. For example, due to public outcry and legal concerns, Google Street View has also added a notification-based approach and has started to publish the routes of its camera cars in some European countries.

We propose *NotiSense*, a simple system that provides useful notifications from urban sensing systems and their participants. These notifications are sent to NotiSense users in the same geographical area where sensing is occurring. NotiSense provides clients with enough information about nearby urban sensing activities such that they become empowered to react appropriately.

In the next section, we describe a typical urban sensing architecture that is assumed by NotiSense. We detail NotiSense's design in section 3 and evaluate a prototype implementation in section 4. We outline remaining challenges and conclude in sections 5 and 6, respectively.

2. URBAN SENSING SYSTEM MODEL

We consider a general urban sensing system model, similar to the *AnonySense* architecture [2]. Our model, shown in Figure 1, consists of three entities: (1) a pool of participating mobile nodes, (2) the urban sensing applications, (3) the urban sensing administration, belonging to an urban sensing service. Mobile nodes are typically consumer-grade mobile devices with built-in sensing components such as GPS receivers, Wi-Fi radios, microphones, cameras, accelerometers, and auxiliary sensing modules connected to an available interface (such as Bluetooth). The responsibilities of the mobile nodes include initiating registration, requesting sensing tasks, performing sensing activities as directed by the tasks, and submitting reports containing collected data. Urban sensing applications submit sensing tasks and request the completed reports generated by the mobile nodes. The urban sensing administration handles registering mobile nodes, assessing the validity of submitted sensing tasks, building a directory of valid sensing tasks for mobile nodes to query, processing reports from the mobile nodes and making the data available to the applications.

3. DESIGN

We list a set of goals for NotiSense, and later use them to evaluate our design.

3.1 Design Goals

- **G1**: Users who opt in ("NotiSense clients") will receive useful sensing notifications based on their preferences.
- **G2**: NotiSense clients should not need to reveal private information such as their fine-grained locations.

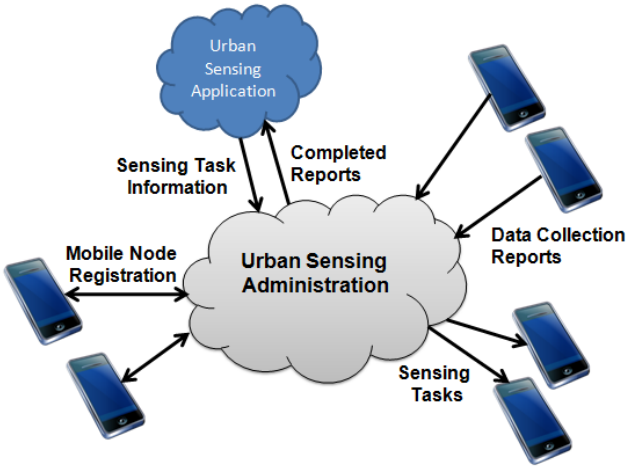

Figure 1: Generalized Urban Sensing Architecture.

- **G3**: Urban sensing participants should not need to publish private information to provide notifications.
- **G4**: The system should have reasonable computation, storage and bandwidth requirements.

G1 is an essential goal as it describes the behaviour of the notification system at a high level. G2 is relevant since NotiSense clients should be strictly gaining privacy, and not losing privacy. Systems like AnonySense [2] are concerned with protecting the privacy of urban sensing participants. We acknowledge that this is a relevant concern for urban sensing in some scenarios, and we feel G3 contributes to a good balance between the privacy of urban sensing participants and bystanders. We would like to design a system that is feasible to implement with today's mobile device hardware and network infrastructure, which is why we include G4.

3.2 Sensing Metadata

Here we describe the types of data that are necessary to make notifications meaningful for NotiSense clients. For each sensing task we need, at the very least, an indication of a geographic area where sensing is requested, an indicator of the types of sensing, and a start and stop time that defines when this information is valid. Supplemental information helps NotiSense clients decide how to respond when they receive a notification. This includes a human-readable description of the urban sensing activity, an indicator of how the collected data will be used, information that describes the movement behaviour of the sensors, and what entity will own the collected data. We assume that this sensing metadata is made publicly available by urban sensing systems.

In addition to information about the sensing task, location information indicating where the data collection is occurring would be extremely useful for someone who is concerned about their privacy. Some urban sensing tasks have specific location requirements, and in others the area is up to the urban sensing participants. Notifications should include the location of the task when it is available, but even more important is the location of the urban sensing participants. It is easy to see that the location privacy of urban sensing participants appears to be conflicting with the ability for NotiSense clients to receive useful notifications. As a com-

promise, the sensing metadata may also include information about a short-range radio broadcast. We discuss this more in section 3.4.

3.3 NotiSense System Architecture

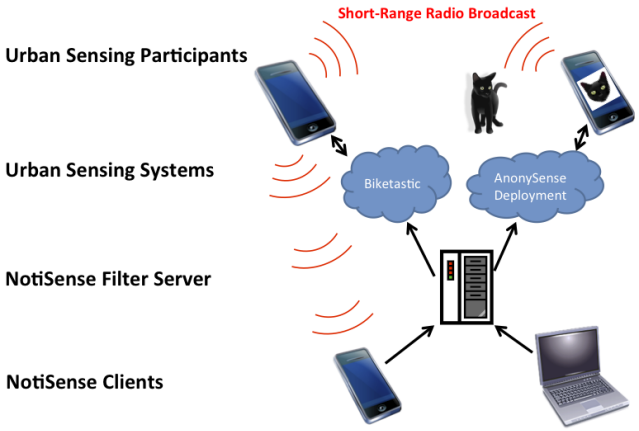

Figure 2: NotiSense's system architecture.

As shown in Figure 2, NotiSense adds three important components to a traditional urban sensing architecture. NotiSense relies on metadata from urban sensing systems, which is collected from urban sensing tasks and consenting urban sensing participants. Since the raw metadata from many urban sensing systems could be too large for mobile devices to periodically download, NotiSense clients rely on filter servers to avoid transferring entire data sets. Finally, some urban sensing participants broadcast short-range wireless signals that are identifiable by NotiSense clients. We discuss below how this architecture can meet our goals.

3.3.1 Filter Servers

Filter servers reduce the amount of sensing metadata downloaded and the number of systems that need to be contacted by NotiSense clients. NotiSense clients express constraints that define the metadata that they are interested in. The filter server will find and return only matching metadata. For example, a client may express that they are only interested in video sensing occurring in their vicinity.

The use of a filter server allows bandwidth and storage requirements to be reasonable, but clients must reveal their locations to the filter server. This is a concern since if a server is able to group requests from a specific user, it may examine this location data to identify that person. Even if we are considering the server to be an honest but curious third party, from the perspective of the NotiSense clients, we do not satisfy G2. We discuss some strategies to accommodate this next.

3.3.2 Protecting NotiSense Client Location Privacy

We consider several approaches to address privacy issues associated with using a filter server. First, the geographical area of interest that a client defines could be adjusted by each NotiSense client. Clients can adjust this to find a privacy level and bandwidth requirements that are reasonable. Second, to make it more difficult to group requests from a specific user, each request can be made through an anonymizing network such as Tor [3]. Third, NotiSense's design is not limited to a single filter server; some individuals or institutions could deploy their own filter servers.

3.4 Short-Range Radio Broadcast

Our third design goal, which states that urban sensing participants should not need to publish private information to provide notifications, appears to conflict with our first goal, which requires the ability to provide useful sensing notifications. It might seem reasonable to take a similar approach to the location privacy of the urban sensing participants as we did above. That is, in the sensing metadata we may only publish coarse-grained location data in a way that is adjustable to a level that the urban sensing participants may be comfortable with. This does in fact hurt our ability to provide useful sensing notifications, especially if we do not have location data for the urban sensing activity. As an example, it is much more difficult to appropriately know how to respond to a notification such as "there are 10 participants in Southern California performing photo sensing for Biketastic". We believe that NotiSense clients need to know when sensing is occurring nearby in order to respond appropriately. We examine how short-range radio transmissions help achieve this.

Many mobile devices today, such as laptops and smartphones, have Bluetooth or Wi-Fi capabilities. A short-range radio broadcast can be used to alert neighbouring bystanders that privacy-impacting sensing is occurring. In order to respect G2, incorporating short-range radios should be passive with respect to the NotiSense clients. That is, the clients should be able to infer that sensing is occurring simply by inspecting short-range radio broadcasts in their area and should not need to establish a two-way connection.

One approach is to encode the sensing metadata in a Wi-Fi Service Set IDentifier (SSID) or a configured Bluetooth device name, which may be passively read. An SSID is limited to 32 alphanumeric characters, and Bluetooth names are often limited to 40 UTF-8 bytes (although the specification allows 248 UTF-8 bytes). We cannot transmit a tuple of all the metadata that we are interested in as described above.

To address this, we include information about a short-range radio broadcast within the published sensing metadata described in section 3.3.2. This information is used by NotiSense clients to determine whether they are in close proximity to the corresponding urban sensing task. We cannot, however, include the Wi-Fi MAC address or the Bluetooth hardware address of a sensing device in the sensing metadata; this information may uniquely identify urban sensing participants or allow an observer to track a participant's location over time, which is once again contrary to G3. Instead, NotiSense assumes urban sensing software may configure an urban sensing participant's device with a random Wi-Fi SSID or Bluetooth name. A cryptographic hash of this value, along with an identifier for the type of radio broadcast, is published by an urban sensing system for a particular urban sensing participant, together with the task's metadata. We publish a cryptographic hash of the random value, instead of the actual value, to avoid trivial false positives that could maliciously be triggered by any user who retrieves these values from urban sensing metadata. The random value will periodically change to further avoid false positives as NotiSense clients discover the value.

Note that the length of the random value used as Wi-Fi SSID/Bluetooth name makes brute-force or dictionary attacks on the cryptographic hash implausible.

As NotiSense clients encounter Wi-Fi or Bluetooth networks in areas that are consistent with the areas in the downloaded sensing metadata, the client will recalculate cryptographic hash values of visible SSID/Bluetooth names. If the calculated hash matches the hash from the downloaded sensing metadata, a notification will be raised based on the user's preferences. If an urban sensing participant chooses to include a coarse-grained location along with the published sensing metadata, or the location of the task they are participating for is available, it will reduce the amount of unnecessary network scanning and hash computations performed by the NotiSense clients.

Our prototype implementation of NotiSense, which uses short-range radio broadcasts, is described in section 4.

A privacy awareness system with similarities to this type of privacy beacon has been studied in the context of ubiquitous computing [9]. PawS is a system that describes how a person can be notified when he enters a ubiquitous computing environment where data is being collected. It describes a mechanism to negotiate a policy, and information about how to access the collected data. The Sensor Tricorder [10] pursues a similar goal. The privacy problem we are considering is more challenging because in an urban sensing setting, we must consider the privacy of the notification issuers, which is not a concern for PawS and the Sensor Tricorder.

4. IMPLEMENTATION

We implemented a prototype NotiSense system with Android 2.3 and Google Nexus One devices. Specifically, we implemented a mock urban sensing participant that generates short-range radio broadcasts as described above, and a NotiSense client to detect these broadcasts.

Android 2.2 implemented the ability to share a cellular data connection to other devices through Wi-Fi (i.e. Wi-Fi HotSpot/tethering). With this, Android can programmatically create a Wi-Fi HotSpot with a specified SSID. This API is not exposed on Android 2.2 or 2.3, but the functionality may be invoked through Java reflection. Our prototype uses this method to implement its short-range radio broadcast.

We also considered using Bluetooth as a short-range wireless broadcast. Android 2.3 exposes the appropriate abilities for mobile applications to make a device discoverable by other Bluetooth devices. However, this will show the user a dialog box warning them that they will be discoverable for the next 60 seconds. This would need to be repeated while the device is broadcasting. This makes using Bluetooth as a privacy beacon as described above not feasible with Android 2.3.

We are particularly interested in the distance within which NotiSense clients can identify radio broadcasts from urban sensing participants. We conducted experiments with our prototype in several locations to learn the effectiveness of using Wi-Fi SSID broadcasts in NotiSense. All experiments used Google Nexus One devices that remained stationary. One device acted as the urban sensing participant and broadcasted an SSID, while the other acted as the NotiSense client. At each measured distance, the NotiSense client scanned Wi-Fi networks every 15 seconds for several minutes and recorded the proportion of scans that detected the urban sensing participant. The NotiSense client only needs to identify an urban sensing participant's SSID once for a notification to be generated. Our experiment gives an idea of the probability that a NotiSense client will successfully identify a nearby sensor at a particular distance in a single scan.

We experimented in three outdoor locations: an open field, a city street, and a university campus. The field had unobstructed line of sight and minimal Wi-Fi traffic with only 6 networks periodically identified during the experiment. The city street had partial line of sight and heavy Wi-Fi traffic (20+ networks consistently identified). On the university campus, the devices were surrounded by buildings, and line of sight was obstructed by part of a building. The Wi-Fi traffic was moderate, with 5 networks persistently identified.

While a photograph may only affect the privacy of individuals in the immediate vicinity, a notification should give NotiSense clients enough time to properly react. Our results, as shown in Figure 3, suggest that this type of Wi-Fi SSID broadcast may achieve a reasonable distance even when there is only partial line-of-sight. In an open field, our Wi-Fi scans reliably yielded the urban sensing participant's SSID at over 300 metres. Our results also suggest that, in an area with many obstructions, NotiSense clients should scan more frequently since the broadcast distance is much more limited.

Power consumption of our implementation was also considered, as NotiSense makes use of Wi-Fi features that are known to consume battery life. We compared the battery drain associated with running the short-range radio broadcast and client Wi-Fi scan operations for an hour to that of an hour of playing media (MP3 music files and video files) on the device. The percentage of the battery power that was drained by an operation was monitored through an Android API. Running the client Wi-fi scan used the least power, consuming 5% of the available battery power over the course of an hour. Executing the short-range Wi-Fi broadcast operation for an hour drained a similar amount of battery power (6%) to playing local MP3 files for the same amount of time. By far the most resource-draining operation we tested was local video play, which drained 18% of the available battery power during one hour.

5. CHALLENGES

A significant challenge for NotiSense is adoption by urban sensing systems. To support short-range radio broadcasts for NotiSense, urban sensing clients and administrative software would need to be modified. Furthermore, mobile devices need to support at least one type of short-range radio broadcast, and their operating systems must allow the customisation of Wi-Fi SSIDs or Bluetooth names by an application.

Urban sensing participants may be unable to use the short-range radio broadcast for another purpose (e.g., the actual sensing) while they are providing notification beacons, as we encountered in our prototype implementation. However, BSMX [14] is a promising method of broadcasting small chunks of data through Wi-Fi beacons without affecting an active Wi-Fi connection. The authors note that the amount of data which may be broadcasted is limited, but they also show that many Android devices have hardware that is capable of BSMX. This is preliminary research, and BSMX is

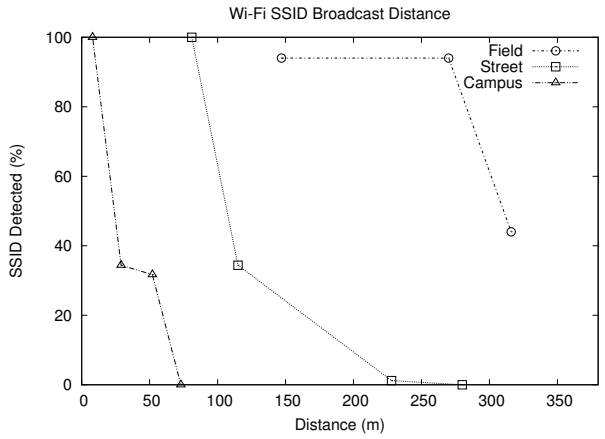

Figure 3: The proportion of Wi-Fi scans from a NotiSense client that successfully identify an SSID broadcast indicating nearby urban sensing.

unlikely to be available through an Android API in the near future.

Some urban sensing tasks may be very short lived. A task could be created and completed in a matter of minutes. This presents a challenge with respect to keeping NotiSense clients up to date. Additionally, NotiSense provides notifications about current potentially privacy-impacting urban sensing tasks. The notifications themselves could be abusive or contain unsolicited advertising.

There certainly could be some scenarios in urban sensing where a short-range radio broadcast is too invasive of the privacy of the urban sensing participants. It should remain at the discretion of participants whether or not they choose to reveal their coarse-grained location or short-range radio broadcast information in order to provide useful sensing notifications to bystanders. In particular, they can choose not to do so if the type of sensing is unlikely to affect a bystander.

We have designed NotiSense with urban sensing as our motivating use case. NotiSense is not limited to urban sensing and could provide useful sensing notifications in other contexts. For example, the data collection procedures for Google Street View and Skyhook [1] could provide NotiSense notifications. Furthermore, one may choose to include sensor metadata for closed-circuit television surveillance systems in public areas (similar to the Sensor Tricorder [10]).

6. CONCLUSIONS

There has been a significant amount of research relating to the privacy of participants in urban sensing systems, but they are not the only parties whose privacy may be impacted. Bystanders who are in an area where sensing is occurring may have some aspect of their state captured without their knowledge. We have proposed NotiSense, a simple system that can provide useful notifications about nearby urban sensing to those who subscribe. Our prototype uses the Wi-Fi SSID as a privacy beacon from urban sensing participants. This can be detected by NotiSense clients without establishing a two-way connection. NotiSense is a simple approach to the issues we have identified, but many challenges remain.

7. REFERENCES

[1] Skyhook. `http://www.skyhookwireless.com/`.

[2] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos. Anonysense: privacy-aware people-centric sensing. In *MobiSys 2008, Breckenridge, CO*.

[3] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *USENIX Security Symposium 2004, San Diego, CA*.

[4] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell. The BikeNet mobile sensing system for cyclist experience mapping. In *SenSys 2007, New York, NY*.

[5] A. Kapadia, D. Kotz, and N. Triandopoulos. Opportunistic sensing: security challenges for the new paradigm. In *COMSNETS 2009, Bangalore, India*.

[6] A. Kapadia, N. Triandopoulos, C. Cornelius, D. Peebles, and D. Kotz. Anonysense: opportunistic and privacy-preserving context collection. In *Pervasive 2008, Sydney, Australia*.

[7] A. Krause, E. Horvitz, A. Kansal, and F. Zhao. Toward community sensing. In *IPSN 2008, St. Louis, MO*.

[8] N. D. Lane, S. B. Eisenman, M. Musolesi, E. Miluzzo, and A. T. Campbell. Urban sensing systems: opportunistic or participatory? In *HotMobile 2008, Napa Valley, CA*.

[9] M. Langheinrich. A privacy awareness system for ubiquitous computing environments. In *UbiComp 2002, Göteborg, Sweden*.

[10] G. Maganis, J. Jaeyeon, T. Kohno, A. Sheth, and D. Wetherall. Sensor Tricorder: What does that sensor know about me? In *HotMobile 2011, Phoenix, AZ*.

[11] S. Murphy et al. Mapping privacy protection in the digital world: Study of the privacy implications of street-level imaging application. `http://publications.gc.ca/collections/collection_2011/parl/XC73-403-1-1-01-eng.pdf`, January 2011.

[12] N. Pham, R. K. Ganti, Y. S. Uddin, S. Nath, and T. F. Abdelzaher. Privacy-preserving reconstruction of multidimensional data maps in vehicular participatory sensing. In *EWSN 2010, Coimbra, Portugal*.

[13] S. Reddy, K. Shilton, G. Denisov, C. Cenizal, D. Estrin, and M. Srivastava. Biketastic: sensing and mapping for better biking. In *CHI 2010, Atlanta, GA*.

[14] S. Schnaufer, S. Kopf, and W. Effelsberg. BSMX – a prototype implementation for distributed aggregation of sensor data. In *PhoneSense 2010, Zurich, Switzerland*.

[15] K. Shilton, J. A. Burke, D. Estrin, R. Govindan, M. Hansen, J. Kang, and M. Mun. Designing the personal data stream: enabling participatory privacy in mobile personal sensing. *Ethics in Science and Engineering National Clearinghouse*, 2009.