Sensibility Testbed: Automated IRB Policy Enforcement in Mobile Research Apps

Yanyan Zhuang
University of Colorado, Colorado Springs

Albert Rafetseder
New York University

Yu Hu
New York University

Yuan Tian
University of Virginia

Justin Cappos
New York University

ABSTRACT

Due to their omnipresence, mobile devices such as smartphones could be tremendously valuable to researchers. However, since research projects can extract data about device owners that could be personal or sensitive, there are substantial privacy concerns. Currently, the only regulation to protect user privacy for research projects is through Institutional Review Boards (IRBs) from researchers' institutions. However, there is no guarantee that researchers will follow the IRB protocol. Even worse, researchers without security expertise might build apps that are vulnerable to attacks.

In this work, we present a platform, Sensibility Testbed, for automated enforcement of the privacy policies set by IRBs. Our platform enforces such policies when a researcher runs code on mobile devices. The enforcement mechanism is a set of obfuscation layers in a secure sandbox, that can be customized for any level of IRB compliance, and can be augmented by policies set by the device owner.

CCS CONCEPTS

• **Security and privacy → Privacy protections**;

KEYWORDS

Privacy protections, Policy enforcement

ACM Reference Format:
Yanyan Zhuang, Albert Rafetseder, Yu Hu, Yuan Tian, and Justin Cappos. 2018. Sensibility Testbed: Automated IRB Policy Enforcement in Mobile Research Apps. In *HotMobile '18: 19th International Workshop on Mobile Computing Systems & Applications, February 12–13, 2018, Tempe , AZ, USA.* ACM, New York, NY, USA, Article 4, 6 pages. https://doi.org/10.1145/3177102.3177120

1 INTRODUCTION

End-user mobile devices, such as smartphones, have become indispensable gadgets in people's everyday lives. As a result, the value of smart devices as data collection vehicles for research studies continues to grow. Since these devices have embedded GPS, accelerometers, cameras, and microphones, they can generate data for large-scale studies such as determining noise levels within an urban neighborhood [9], or studying traffic patterns [15].

However, for device owners, privacy threats to mobile devices have increased dramatically due to these sensors[1] and the sensitive data they generate. Potential attackers seek to take advantage of the rich functionality of sensors on mobile devices. Therefore, Institutional Review Boards (IRBs) review research protocols to evaluate whether researchers collect user data ethically. However, IRBs cannot ensure that a curious or erroneous researcher will follow the protocols for user data. Even worse, attackers might hack into an experiment to steal sensitive data from users [1]. For researchers without security expertise, it is particularly difficult to protect participants' privacy.

We introduce Sensibility Testbed, a testbed that streamlines the process of running IRB-compliant experiments on mobile devices. Sensibility Testbed simplifies the process of implementing IRB-compliant data access policies, without relying on the researcher to protect sensitive data. Instead, it technically enforces a series of policies that limit what information can be collected from end-user devices and how often. Researchers can establish a secure, direct connection with remote mobile devices to run experiments without any policy violation. Furthermore, they do not need to build their own app and deploy in an app store to collect data; they only need to configure IRB policy, and write about one line per sensor to collect data with our platform.

Sensibility Testbed uses policies to define the granularity of access for all sensors conforming to a researchers's IRB, such as identifying the city where they live, without revealing the exact address; accessing GPS every ten minutes, instead of continuously. In addition, Sensibility Testbed has a set of *baseline policies* that are always enforced on each research experiment. These policies address common attacks, and by default disable access to sensors of high risks, such as cameras and microphones. Sensibility Testbed's infrastructure applies the IRB policies specified by the researcher's institution, by implementing the policies on end-user devices. These policies can be customized according to the types of sensors accessed by an experiment. Finally, device owners can also adjust the privacy settings locally through configuring these policies.

All these policies are enforced through *obfuscation layers* in a secure sandbox. Each obfuscation layer is customized to limit the precision of data collected, the frequency with which the sensor can

HotMobile '18, February 12–13, 2018, Tempe , AZ, USA

ACM ISBN 978-1-4503-5630-5/18/02…$15.00
https://doi.org/10.1145/3177102.3177120

[1] We broadly define sensors as hardware that can record phenomena about the physical world, e.g., the WiFi/cellular network, GPS location, movement acceleration, etc.

be accessed, or both. All obfuscation layers are programmable to meet each device owner's privacy preference, and each institution's IRB requirements.

The contributions of this work are as follows:

- We introduce Sensibility Testbed as a platform for experimentation on mobile devices that enables programmable enforcement of IRB policies.
- We develop and integrate a set of baseline privacy policies into the testbed design that respond to common attack techniques identified in the literature. These policies prevent attackers from accessing private data on personal devices.

2 OVERVIEW

In this section, we use an example to show how a researcher uses Sensibility Testbed to conduct an experiment. We assume that Alice, a device owner, participates in a research experiment, while a researcher, Rhonda, wants to run code using a number of devices, including Alice's.

Interaction among different parties. Alice decides to install the Sensibility Testbed app because she altruistically wants to help scientific progress. She may configure the privacy settings in her app, e.g., to block any possible access to her microphone. Once the app is started, an instance of the testbed sandbox (Section 4.1) will be created. At this point, her device is ready for researchers to use.

Rhonda wants to study different cellular technologies in her city. She wants to gather the network type (3G, 4G, LTE, etc.), provider, and signal strength from device owners. Rhonda registers her experiment with Sensibility Testbed (Section 3.3) and enters information about the types of data her experiment requires. Rhonda's IRB protocol specifies that she requires accurate carrier network information, such as cellular signal strength, network type, but randomized cell IDs in lieu of the cell ID that the device is associated with. She also configures that her experiment requires GPS data to be within 30 meters of accuracy for her measurements, and needs to update this information every 10 minutes.

If Rhonda's IRB protocol requests access to sensors in a manner that is equal to or at a coarser level than Sensibility Testbed's baseline policies, her experiment will be immediately approved. If not, Rhonda's experiment will be subject to an additional check at Sensibility Testbed. If approved, Rhonda can deploy her experiment on remote end-user devices (including Alice's), which she can request through Sensibility Testbed. She may start/stop her experiment at any time, and collect the results from the remote devices using an `ssh`-like console. Even if attacker Eve hacks into Rhonda's experiment, the sensor access will still be blocked except in the manner specified by the IRB and baseline policies.

Threat model. We assume that Rhonda may inadvertently access private data on Alice's devices. However, Rhonda's IRB cannot prevent this because IRB does not know Rhonda's implementation details. Furthermore, an attacker may maliciously compromise Rhonda's experiment to collect data. Sensibility Testbed provides protection against all these threats.

3 TESTBED DESIGN

In this section, we use the example in Section 2 to explain the design of Sensibility Testbed. We first present an overview of the requirements to build a smartphone testbed, and then describe the detailed design as Rhonda studies the cellular service quality using Alice's device.

3.1 Testbed Requirements

To design and implement a mobile testbed, we have to strike a balance between the security guarantee of code execution, the privacy of device owners, and usability including the programming interface and experiment setup. We summarize the main requirements as follows.

Security Guarantee of Code Execution. To responsibly provide access to smartphones, a smartphone testbed should provide security guarantee for the experiments running on the device. Any experiment should not affect a device owner's normal interaction with the other apps, and should never do any damage to the device's file system, slow down network connectivity, etc.

Privacy Protection and IRB Compliance. An equally important requirement is to provide privacy protections for device owners, and ensure that experiments comply to IRB policies that involve privacy. This prevents researchers from accidentally over gathering data, and further enables a wide range of research that were difficult to perform due to the overhead of IRB.

Informed Consent is one of the foundations of responsible research. It involves having participants understand the overarching goals, procedures, and risks of the research that will be performed on them (or using their data) and for them to indicate their willingness to participate. Appropriate materials must be provided in lay language so that participants can comprehend what they agree to.

Usability. Last but not least, it is crucial to make it easy for researchers to access the testbed and deploy experiments on a variety of devices. This requires a user-friendly interface, as well as a well-designed and well-managed infrastructure.

In the following, we present the design choices we made according to the requirements above.

3.2 Informed Consent

In designing Sensibility Testbed, we note that informed consent need not be done individually for every use of data. It is common in medical research, social sciences, economics, and other fields to simply have participants opt in for their data to be used for research purposes related to that field, especially in cases where the research involves low or no risk for participants. We have already obtained IRB approval for Sensibility Testbed to use a similar structure, where participants opt in to computer science research with low to no risk for participants. Therefore, Alice consents to provide access to researchers like Rhonda to run IRB-approved experiments on her devices. Meanwhile, Rhonda is bound to the IRB agreement of her institution, and is also bound by the policies of Sensibility Testbed.

3.2.1 Device Owner Policy. Also part of informed consent is that device owners can control how information is gathered from their devices. For example, Alice can opt out of individual experiments, disable or stop all experiments at any time. Furthermore, she can control, in a more precise manner, how sensors are accessed on her device, in a way she is comfortable with. The device owner's policies supersede any policies set by researcher's IRB. For example, if Alice disallows access to her microphone, then Rhonda's experiment

Privacy concerns	Sensor data	Baseline policies†
Low risk.	Battery status (charging/discharging), temperature, technology, health (good/overheat), battery level, voltage, plug-in type.	Full precision, round-up (if numeric), or constant.
	Bluetooth scan mode, state (enabled/disabled).	
	Cellular network roaming status, SIM card status (ready/absent), phone status (idle/busy), signal strength.	
	Location service provider.	
	WiFi link speed, association state, nearby routers' frequency, signal strength.	
	Vibrate mode, screen settings (on/off, brightness, timeout), media/ringer volume.	
Prevent keyloggers and activity tracking.	Motion sensors: accelerometer, gyroscope, magnetometer, orientation, ambient light, etc.	Full precision, round-up, random rotation, constant; restrict access frequency.
Prevent locating a device.	Latitude, longitude, altitude.	Approximate to the nearest zipcode region, or city/state/country center; restrict access frequency.
	Nearby Bluetooth device names.	Hashed device names; restrict access frequency.
	Cellular network cell ID, neighboring cell ID(s).	Randomized ID; restrict access frequency.
	Cellular network operator ID and name, country code, area code.	Hashed ID, names, and code; restrict access frequency.
	WiFi connection information (SSID and MAC address of currently connected router).	Hashed SSID, randomized MAC address; restricted access frequency.
	WiFi scan result (nearby WiFi routers' SSIDs and MAC addresses)	
Prevent identifying a device owner.	Bluetooth MAC address, local name.	Randomized MAC address, hashed device names.
	Cellular device ID, incoming number.	Randomized ID and number.
	WiFi connection information (device MAC address, IP address).	Randomized MAC address, hashed IP address.
Prevent video/audio recording.	Take pictures, record videos using a camera.	Disabled.
	Voice record using a microphone.	
Prevent actions for owner.	Scan barcode, search, etc., using an Intent.	
	Send/receive messages, delete messages, dial/pick up phone calls.	
Protect owner's contacts.	Contact list of the device owner in an address book.	

†This lists the policies at publication time. Policies need to be adjusted as new threats emerge.

Table 1: Sensibility Testbed's baseline policies for sensor data.

cannot get access to Alice's microphone, even if the IRB policy at Rhonda's institution allows the access.

3.3 Researcher Specifies IRB Policies

Before conducting any experiments, Rhonda first registers an account with Sensibility Testbed. The testbed sets up the relevant IRB policies that must be enforced on remote devices on behalf of Rhonda.

To register an experiment, Rhonda must indicate the precision and frequency at which Rhonda's IRB protocol allows her experiment to access each sensor. The list of sensors and their available policies defined by Sensibility Testbed are in Table 1. Each policy can be further customized. Rhonda sets the policies by specifying that her experiment can (1) read location information from devices with accuracy within 30 meters, (2) read accurate cellular signal strength and network type, but use randomized cell IDs, and (3) get location and cellular network updates every 10 minutes. Lastly, she specifies the experiment duration, after which the testbed deletes her experiment. All the information above is checked against the approved IRB certificate to ensure that the policies are consistent. This information is used to define obfuscation layers that enforce technical restrictions for her experiment. These obfuscation layers cannot be bypassed.

When an account is approved, Rhonda can request a number of mobile devices for her experiment through the testbed. If Alice's device (among other devices) is discovered by the testbed, Sensibility Testbed assigns a sandbox on her device to Rhonda's account, in which Rhonda's experiment will run. The testbed then creates access policies for Rhonda's experiment in accordance with her specified IRB policies, and deploys them on Alice's device.

3.3.1 Baseline Policies. Sensibility Testbed uses *baseline policies* to prevent common privacy and security attacks, as listed in Table 1. The baseline policies disable highly sensitive sensors, such as cameras and microphones. Additionally, the baseline policies disable intrusive actions such as making phone calls, scanning a barcode on behalf of the device owner, or accessing an address book, and so on.

Furthermore, the baseline policies obfuscate some common privacy risks: (1) identifying a device or its owner via, e.g., MAC address and device ID; (2) locating a device through cell IDs, WiFi SSIDs, etc.; and (3) inferring keys strokes and activities of a device owner using motion sensors such as accelerometers and gyroscopes. For example, the policies enforce randomized MAC addresses in a Bluetooth or WiFi network, and approximated location coordinates, as well as controlling the frequency of access to motion sensors. Note that keyloggers and activity trackers are more effective when the access frequency to motion sensors is high [10, 14].

Finally, there are low-entropy sensors like battery status and WiFi link speed whose privacy impact is small. Access to them still must be requested through the IRB process, as the goal is to minimize the privacy risk for device owners whenever possible.

Sensibility Testbed's baseline policies are set to appropriate levels to protect against known attacks today. However, these levels will need to change over time as new attacks emerge, making the baseline policies stronger over time.

3.3.2 Policy Hierarchy. Device owner's policies are always applied first. Following this, the baseline policies for the experiment are used. Then the experiment-specific IRB policies are put into place. The experiment code is subject to all policies. The ability to combine policies makes it easy to create complex policies. Next, we introduce how these policies are implemented.

4 IMPLEMENTATION

We describe the implementation details of Sensibility Testbed, including its secure sandbox, and the way to enforce privacy policies as described in Section 3.

4.1 Secure Sandbox

The sandbox in Sensibility Testbed provides a programming language interface equipped with system calls for networking, file system, threading, locking, logging, and most importantly, sensors. Every system call is strictly sanitized to preserve consistent behavior across different OSes, and to avoid exploitable vulnerabilities. Additionally, the sandbox can interpose on system calls that use resources, such as network, disk I/O, and sensors, and prevents or delays the execution of these calls if they exceed their configured quota. The details of the sandbox implementation can be found in our prior work [3].

4.2 Policy Enforcement

Rhonda's IRB policies are implemented as obfuscation layers, with each layer enforcing an access control policy over a sensor. The sandbox provides a list of system calls, such as `get_location()`, `get_accelerometer()`. The sandbox can also control the behavior of these calls using system call interposition [3]. Each obfuscation layer is thus implemented as template code, pre-loaded in each sandbox, and can be instantiated with parameters from Rhonda's policy specification to interject a system call. Each obfuscation layer defines one or two of the following categories of policies.

4.2.1 Reducing Data Precision. As an example, to obfuscate Alice's location to a nearest city, the obfuscation layer conceptually takes these steps:

```
define get_city_location():
  # call the full-precision get_location function
  exact_location = get_location()

  # look up city corresponding to the exact location
  city_location = find_closest_city(exact_location)

  return city_location
```

The function first retrieves the device location at full precision. It then returns the closest city as the obfuscated device location. In every sandbox that Rhonda can access, each function call to `get_location` is interposed and then replaced by a call to `get_city_location`. This is achieved in a transparent and non-bypassable way, much like a derived class can override its parent class's implementation of a method.

4.2.2 Restricting Data Access Frequency. When an experiment attempts to use a sensor more frequently than the given threshold allows, the obfuscation layer pauses the code for as long as required to bound it, on average, below the threshold. The code of a rate-limiting obfuscation layer for accessing an accelerometer is as follows.

```
define rate_limited_accelerometer(pause_time):
  # pause the code for a time threshold
  lock.acquire()

  while (current_time < next_allowed_access_time):
    pause

  # update time threshold for next sensor access
  next_allowed_access_time = \
      current_time + pause_time
  lock.release()

  # call the original get_accelerometer function
  accelerometer = get_accelerometer()
  return accelerometer
```

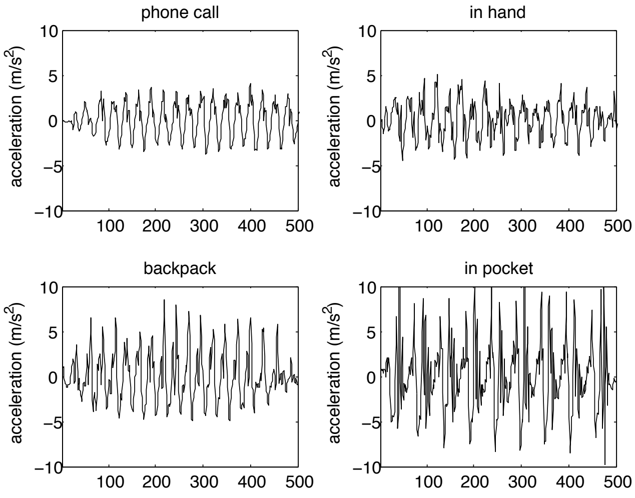

Figure 1: Accelerometer data based on different activities.

The rate-limiting code ensures that enough time has elapsed before the accelerometer is accessed. Note that locking is necessary to prevent race conditions among different threads that try to access the sensor at the same time. Access frequency is controlled by `pause_time`, a parameter determined by Rhonda's IRB policies. After enough time has elapsed, the code accesses the accelerometer, and returns its reading. When this obfuscation layer is in place, all calls to `get_accelerometer()` will be replaced by `rate_limited_accelerometer(pause_time)`.

4.2.3 Policy Stack. Different sets of policies can be customized as a *policy stack*. In this stack, every layer inherits the policy defined by its ancestor layers, with the exception of the lowest layer (the sandbox kernel). The experiment runs at the top of the policy stack, inheriting all the policies defined by the lower layers. Each policy stack acts as a set of filters for different sensors, through which sensor calls must pass before being accessed by a sandboxed program.

5 EVALUATION

The mobile testbed in this work serves two purposes: providing resources for conducting research experiments, and protecting end-users' privacy. Since it is difficult to evaluate the Sensibility Testbed IRB workflow from a researcher's point of view, we discuss technical issues as well as the system's usability and practicality.

Q1: Do the proposed privacy policies effectively protect device owners in research experiments? Here we provide an example where an adversary uses an accelerometer to infer a device owner's everyday activities. We show that a Sensibility Testbed policy can effectively prohibit such activity tracking.

Activity tracking. A user's activities, such as reading text messages or emails, making a phone call, or carrying the phone in a backpack, are reflected by motion sensors. These sensors do not require permission from the device owner to access, therefore attacks using them are more difficult to detect and prevent. Figure 1 shows the accelerometer data patterns of a device owner when the access rate is 50 Hz. The x-axis is the number of samples, and the y-axis is the accelerometer data with the gravity removed. From this figure, a device owner's activities can be inferred by detecting periodic maxima in the magnitude of acceleration. These maxima can also be used to segment the data into individual steps, which provide a signature that is unique to each device owner and the specific activity. For example, when the device is placed in the owner's pants pocket, for each pair of steps, the data will show a large spike and a smaller spike due to leg swings. Nevertheless, when the signal is reduced to below 25 Hz, every pattern that we saw looks like the one when the device is held in hand.

Policy enforcement. To test the effectiveness of policy enforcement, we recruited 16 device owners and asked them to carry their phones in each of the four modes. The raw accelerometer data rate from their devices varied from 50 to 100 Hz. We subsampled the raw accelerometer data with rates from 20 to 50 Hz. Each activity's tracking accuracy with different access rates are shown in Figure 2.

There is a sharp decline in the tracking accuracy when the access rate drops below 25 Hz. At this rate, it is impossible to distinguish the activity of the device owner, as all patterns become similar to the reference pattern when the device is held in hand. Thus, in Figure 2 the accuracy when the phone is held in hand is always greater than zero. To prevent activity tracking, Sensibility Testbed sets a baseline policy to restrict motion sensor's frequency to below 25 Hz.

Q2: What utility does restricted data provide? We showed above that privacy policies effectively prohibit tracking a user's activities. However, the rate-limited data still suffices for many other applications, e.g., the accelerometer data reduced to 25 Hz can be used for pedometry, as one can still differentiate one step from another in the accelerometer data.

In another experiment, Sensibility Testbed was used by a high school student as part of a vehicle data collection project. The student connected his device to the on-board diagnostics (OBD) sensor interface in a car, and used Sensibility Testbed to capture data, such as fuel consumption, pressure, mileage, and engine RPMs. The student then drove around the NYC area and used this data to derive information about traffic patterns [13]. Even when the location was restricted to a ZIP code area, the data allowed him to make inferences about traffic conditions, when combined with information about the weather or large gatherings at entertainment venues, and to predict possibly hazardous road conditions.

While it is an open question that whether the policies using obfuscation may affect the accuracy of an experiment, we plan to carry out more studies like the ones above, to investigate such a privacy-functionality tradeoff.

Q3: Is Sensibility Testbed effective for developing sensor-enabled experiments? In the past, we have hosted hack-a-thons co-located with the IEEE Sensors Applications Symposium (SAS) [2]. This conference attracts a diverse community of researchers that use sensors in their research. The vast majority of participants come from other scientific disciplines than computer science. Each time we have had about twenty participants spend a day of the conference building applications using Sensibility Testbed. None of the participants had any prior experience with our platform.

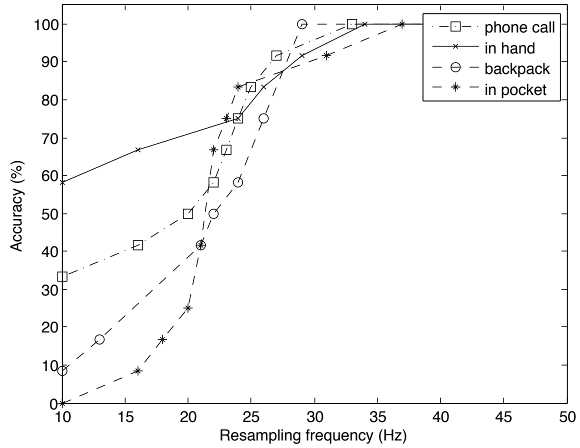

Figure 2: Accuracy at different resampling frequencies.

Researchers implemented code that they tested in the same day. Despite only knowing about Sensibility Testbed for roughly six hours, many researchers built many interesting and complex applications. These included applications for navigating between conference rooms using WiFi connection information, and monitoring battery information and turning off WiFi and Bluetooth when battery level is low. Among all the 25 teams we had, only one group did not finish the application development.

6 CHALLENGES

During the implementation and evaluation of Sensibility Testbed, we observe the following research challenges, which we will leave to our future work.

6.1 Accessing Personal Data

Due to privacy considerations, Sensibility Testbed deliberately disallows access to personal data stored on the phone (such as performing research using the device owner's phone book), or introspection into the mobile OS. We consider the obfuscation techniques presented in this paper valid for these resources as well. However, interesting approaches for querying privacy-preserving database would be helpful to improve the utility of the testbed.

6.2 Technical Challenges

Sensibility Testbed is designed to minimize the privacy repercussions of smartphone research by limiting an experiment's access to sensors. However, this does not guarantee that the existing obfuscation layers and configurations will always be able to address adequately all possible privacy concerns. Therefore, we anticipate that the baseline policies will need to adapt over time. Additionally, bugs in the sandbox or obfuscation code might expose the device owner's privacy, thus requiring updates of the platform code.

6.3 Usability Challenges

Letting device owners choose and parametrize their own privacy policies also poses a usability challenge. There are many interesting problems to explore in this space. For example, how to help device owners configure the policies to make informed decisions about how their devices are used, whether device owners would want to have different policies to different researchers, and so on.

7 RELATED WORK

Sensibility Testbed is the first mobile testbed that supports automatic IRB policy enforcement. We compare our work with previous research in this section.

7.1 Data Anonymization

Some researchers employ a third-party anonymizing agent as a proxy between the data source and the service using the anonymized data [6, 11]. Sensibility Testbed's privacy preservation is carried out on the device, without using a third-party agent. The only data leaving the device is what researcher requests through the IRB policies.

7.2 Data Obfuscation

Data obfuscation has been suggested in [8] for privacy preservation. The authors demonstrate location and time obfuscation of reports. Sensibility Testbed built a systemic solution for data obfuscation, and enable purpose-based access control for the data (eg., the granularity of data-sharing depends on the utility), which is never studied systemically in previous papers. Besides, we add new schemes such as topological obfuscation (e.g. mapping exact locations to ZIP code areas), hashing, and randomization, and also applies them to other sensor types.

7.3 Detecting Privacy Violation

There has also been much work dedicated to detecting privacy violation from mobile apps [4, 5, 7, 16]. These approaches alert the user when sensitive data is exfiltrated from the device, either at runtime [4, 5] or install time [7]. Although these systems notify the user when there is a potential privacy breach, they leave the mitigation decision up to the user because they do not know whether the data sharing is legitimate or not. Sensibility Testbed, on the other hand, protects the user directly from exfiltration of sensitive data without requiring manual intervention at a critical time, because Sensibility Testbed conforms to the IRB protocol.

7.4 Other Smartphone Testbed

There are several smartphone testbeds with researcher purchased devices. For example, PhoneLab [12] is a smartphone testbed for research experimentation. It provides low-cost devices to students. However, it only requires researchers to submit their approved IRB protocols and the URLs to their apps in the Google Play Store. This model cannot enforce privacy policies as Sensibility Testbed does.

8 CONCLUSION

By enabling programmable enforcement of IRB-approved privacy policies through the control of sensor access, Sensibility Testbed is able to provide flexible policy implementation. As a result, not only researchers no longer need to have a "hands-on" role in policy enforcement, it also greatly reduces the risk of participation for device owners. Sensibility Testbed thus encourages the development of larger-scale experiments, protects the privacy of participants, and is easy for researchers to use.

REFERENCES

[1] [n. d.]. Hackers can attack your phone via 76 popular iOS apps. ([n. d.]). Accessed January 14, 2018, http://www.dailymail.co.uk/sciencetech/article-4203180/Hackers-attack-phone-76-popular-iOS-apps.html.

[2] [n. d.]. Sensors Applications Symposium. ([n. d.]). http://sensorapps.org/.

[3] Justin Cappos, Armon Dadgar, Jeff Rasley, Justin Samuel, Ivan Beschastnikh, Cosmin Barsan, Arvind Krishnamurthy, and Thomas Anderson. 2010. Retaining sandbox containment despite bugs in privileged memory-safe code. In *Proceedings of the 17th ACM conference on Computer and communications security.* ACM, 212–223.

[4] Supriyo Chakraborty, Chenguang Shen, Kasturi Rangan Raghavan, Yasser Shoukry, Matt Millar, and Mani Srivastava. 2014. ipShield: a framework for enforcing context-aware privacy. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14).* USENIX Association, 143–156.

[5] William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N Sheth. 2014. TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)* 32, 2 (2014), 5.

[6] Marco Gruteser and Dirk Grunwald. 2003. Anonymous usage of location-based services through spatial and temporal cloaking. In *Proceedings of the 1st international conference on Mobile systems, applications and services.* ACM, 31–42.

[7] Shashank Holavanalli, Don Manuel, Vishwas Nanjundaswamy, Brian Rosenberg, Feng Shen, Steven Y Ko, and Lukasz Ziarek. 2013. Flow permissions for android. In *Automated Software Engineering (ASE), 2013 IEEE/ACM 28th International Conference on.* IEEE, 652–657.

[8] Apu Kapadia, Nikos Triandopoulos, Cory Cornelius, Daniel Peebles, and David Kotz. 2008. AnonySense: Opportunistic and privacy-preserving context collection. In *Pervasive Computing.* Springer, 280–297.

[9] Chucri A Kardous and Peter B Shaw. 2014. Evaluation of smartphone sound measurement applicationsa). *The Journal of the Acoustical Society of America* 135, 4 (2014), EL186–EL192.

[10] Emiliano Miluzzo, Alexander Varshavsky, Suhrid Balakrishnan, and Romit Roy Choudhury. 2012. Tapprints: your finger taps have fingerprints. In *Proceedings of the 10th international conference on Mobile systems, applications, and services.* ACM, 323–336.

[11] Mohamed F Mokbel, Chi-Yin Chow, and Walid G Aref. 2006. The new Casper: query processing for location services without compromising privacy. In *Proceedings of the 32nd international conference on Very large data bases.* VLDB Endowment, 763–774.

[12] Anandatirtha Nandugudi, Anudipa Maiti, Taeyeon Ki, Fatih Bulut, Murat Demirbas, Tevfik Kosar, Chunming Qiao, Steven Y Ko, and Geoffrey Challen. 2013. Phonelab: A large programmable smartphone testbed. In *Proceedings of First International Workshop on Sensing and Big Data Mining.* ACM, 1–6.

[13] Michael Reininger, Seth Miller, Yanyan Zhuang, and Justin Cappos. 2015. A First Look at Vehicle Data Collection via Smartphone Sensors. In *Sensors Applications Symposium (SAS), 2015 IEEE.* IEEE.

[14] Zhi Xu, Kun Bai, and Sencun Zhu. 2012. Taplogger: Inferring user inputs on smartphone touchscreens using on-board motion sensors. In *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks.* ACM, 113–124.

[15] Yanyan Zhuang, Jianping Pan, Yuanqian Luo, and Lin Cai. 2011. Time and location-critical emergency message dissemination for vehicular ad-hoc networks. *Selected Areas in Communications, IEEE Journal on* 29, 1 (2011), 187–196.

[16] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven M Bellovin, and Joel Reidenberg. 2017. Automated analysis of privacy requirements for mobile apps. In *Proceedings of the Network and Distributed System Security (NDSS) Symposium*, Vol. 2017.