UNDERSTANDING AND CAPTURING PEOPLE’S MOBILE APP PRIVACY PREFERENCES

Jialiu Lin

CMU-CS-13-127
October 28, 2013

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Norman Sadeh, Co-Chair
Jason I. Hong, Co-Chair
Mahadev Satyanarayanan
Sunny Consolvo, Google

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

This research was supported in part by the National Science Foundation under grants CNS-1012763, NSF CNS-1228813 and CNS-0905562, in part by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273 from the Army Research Office, in part by NSA under grant W911NF0910273, in part by Carnegie Mellon Portugal ICTI 1030348 and in part by Google.

Keywords: mobile app privacy, app analysis, expectation, privacy interface, user studies, privacy preferences, crowdsourcing, user-oriented machine learning, Android permission

Abstract

Users are increasingly expected to manage a wide range of security and privacy settings. An important example of this trend is the variety of users might be called upon to review permissions when they download mobile apps. Experiments have shown that most users struggle with reviewing these permissions. Earlier research efforts in this area have primarily focused on protecting users' privacy and security through the development of analysis tools and extensions intended to further increase the level of control provided to users with little regard for human factor considerations.

This thesis aims to address this gap through the study of user mobile app privacy preferences with the dual objective of both simplifying and enhancing mobile app privacy decision interfaces. Specifically, we combine static code analysis, crowdsourcing and machine learning techniques to elicit people's mobile app privacy preferences. We show how the resulting preference models can inform the design of interfaces that offer the promise of alleviating user burden when it comes to reviewing the permissions requested by mobile apps. Our contribution is threefold. First, we provide the first large-scale, in-depth analysis of mobile app data collection and usage practices as found in the Google Play app store. This includes an analysis of over 100,000 Android apps, the permissions they request and the different types of third parties with which they share information. Second, we introduce a crowdsourcing methodology to collect people's privacy preferences when it comes to granting permissions to mobile apps for different purposes (e.g. for internal purpose, for sharing with advertising networks) and use the results to develop new mobile app privacy decision interfaces. Third, by using machine learning techniques to analyze privacy preferences from over 700 smartphone users, we show that, while these preferences are diverse, a relatively small number of privacy profiles can go a long way in simplifying the number of decisions users have to make. This last contribution offers the promise of alleviating user burden and ultimately increasing their control over their information.

This thesis provides an important scientific basis for starting to reconcile mobile privacy and usability and, in particular, helping inform the design of more usable privacy interfaces and settings.

Acknowledgements

I thank Prof. Jason I. Hong and Prof. Norman Sadeh for advising me on this work. Their expertise and insights greatly enriched my work. My committee members, Prof. Mahadev Satyanarayanan and Dr. Sunny Consolvo have offered constructive questions and comments, which have greatly helped me improve this thesis work. I also thank other faculty and fellow students who provided valuable feedback on my thesis, in particular Prof. Janne Lindqvist, Prof. Joy Zhang, Shahriyar Amini, Richmond Wong, Prateek Sachdeva, Rajarshi Das, Dr. Nan Li, Bin Liu, Jason Wiese, and Eiji Hayashi. In addition, I would like to thank my family for their continuous support during my arduous and joyful journey of pursuing my PhD degree.

This research was supported in part by the National Science Foundation under grants CNS-1012763, NSF CNS-1228813 and CNS-0905562, in part by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and W911NF-09-1-0273 from the Army Research Office, in part by NSA under grant W911NF0910273, in part by Carnegie Mellon Portugal ICTI 1030348 and in part by Google.

Table of Contents

Abstract iii

Acknowledgements iv

1 INTRODUCTION 1

1.1 Overview 1

1.2 Three Major Components of this Thesis 3

1.2.1 Analyzing Apps’ Privacy-Related Behavior 3

1.2.2 Improving Ways of Notifying Users 4

1.2.3 Helping Users with Privacy Settings by Providing Privacy Profiles 4

1.3 Research Contributions and Future Prospects 4

2 BACKGROUND AND RELATED WORK 6

2.1 Android & iOS Privacy Frameworks 6

2.2 Security-oriented Approaches in Mobile App Privacy 8

2.2.1 Permission analysis 8

2.2.2 Static analysis 9

2.2.3 Dynamic analysis 10

2.2.4 Security and privacy extensions 11

2.3 End-User Research in Mobile App Privacy 11

2.4 User Modeling in Location Sharing 13

2.5 Crowdsourcing and Human Computation 15

2.6 Relationship to Prior Work 15

3 DISSECTING AND UNDERSTANDING THE BEHAVIOR OF SMARTPHONE APPS 17

3.1 Data Gathering 17

3.2 Dissecting Android Apps 17

3.3 Analysis Results 19

3.4 Other Potential Ways to Analyze Apps 24

3.5 Summary 25

4 IDENTIFYING PATTERNS IN APPS’ PRIVACY BEHAVIORS 27

4.1 Preprocessing 27

4.2 Clustering Algorithms and Distance Functions 27

4.3 Evaluating Clustering Algorithms 29

4.4 Resulting Clusters 32

4.4.1 Cluster_1: Few requested permissions 33
4.4.2 Cluster_2: Permissions primarily requested for internal use 34
4.4.3 Cluster_3: Ad-powered apps 34
4.4.4 Cluster_4: Apps that promote sharing 36
4.4.5 Cluster_5: Multi-purpose apps 37
4.5 Discussion 37
5 NOTICE & AWARENESS: HOW TO INFORM USERS? 39
5.1 What to Show? 39
5.2 How to Gather Data? 40
5.3 Study Description 41
5.3.1 Feasibility of Using Crowdsourcing to Study Privacy 42
5.3.2 How Users Feel about Popular Apps 43
5.3.3 Expectation, Purpose, and Comfort Level 43
5.4 How to Apply the Results? Preliminary Design of a New Privacy Summary Interface 44
5.5 Privacy Interfaces With Different Layouts 47
5.5.1 Proposed Privacy Interfaces 47
5.5.2 Evaluation Methodology 50
5.5.3 High Level Results 51
5.6 Summary 53
6 CHOICES & CONTROLS: HOW TO HELP USERS WITH THEIR PRIVACY SETTINGS? 55
6.1 Usability Challenges in Managing Privacy Settings 55
6.2 Crowdsourcing Study 55
6.3 Users' Average Preferences and Their Variances 56
6.4 Clustering on Users' Preferences 58
6.4.1 Algorithms and Clustering Results 58
6.4.2 Making Sense of Privacy Profiles 60
6.5 Implications of Privacy Profiles 62
6.5.1 Privacy profiles and demographic information 62
6.5.2 Identifying privacy profiles: what questions to ask 65
6.5.3 Other potential applications 68
7 CONCLUSION AND FUTURE WORK 70
7.1 Thesis Summary 70
7.2 Future Work 71

7.2.1 Leverage NLP techniques to further understand the functionality of the app 71

7.2.2 User studies to evaluate identified privacy profiles.................................... 71

7.2.3 Design, implement and deploy a privacy wizard.. 71

8 REFERENCES 72

Appendix. A…………………………………………………………………………...79

Appendix. B…………………………………………………………………………...80

1 INTRODUCTION

1.1 Overview

Smartphone ownership has grown rapidly over the last few years. In 2013, global smartphone shipments are expected to hit 1 billion units [64]. Nearly half of cell phone owners carry smartphone nowadays. The explosion in smartphone ownership has been accompanied by the emergence of App Stores that enable users to download a growing number of applications onto their devices. As of June 2013, the Google Play Store[1] offered more than 1,000,000 apps; the Apple App store offered more than 950,000 apps, and both with close to 50 billion downloads since its launch [117, 118]. Mobile apps can make use of numerous capabilities of a smartphone, such as a user's current location and call logs, providing users with pertinent services and attractive features**.**

Inevitably, access to these capabilities opens the door to new types of security and privacy intrusions. Malware is an obvious problem [33, 52]; another serious problem is that mobile users, in general, are neither fully aware of nor have full control over how mobile apps access and transmit personal information. For example, the Pandora music app was under federal investigation for gathering location data, gender, year of birth, and unique device ID from mobile users and sharing this information with advertisers [35]. Social network applications, such as Facebook and Path, were found uploading entire contact lists onto their servers, which greatly surprised users and made them feel very uncomfortable [63, 111]. In fact, studies [54, 77, 82] have shown that users have a poor understanding of these sensitive resource usages, and existing interfaces fall short in terms of providing users with the information necessary to make informed decisions.

A number of ongoing research efforts focus on protecting mobile users' privacy and security using software analysis techniques or security extensions with app-specific privacy controls (e.g., [29, 68, 123]). In Android 4.3, Google also released a hidden "App Ops" function which allows users to fine-tune their permission settings after installation [114]. Given the average number of apps users install and the average number of permissions each app requests, asking users to systematically configure all these settings seems unrealistic. It creates too great a burden on the users and would most likely overwhelm users with details they may not fully understand and may ultimately not care about. To date, though there is a handful of work approaching the mobile app privacy problem from the users' perspective [51, 54, 77], little work has been done to understand people's privacy preferences in using mobile apps and see to what extent a better understanding of these preferences could inform the design of interfaces that empower users to better manage their privacy.

The fundamental goal of this thesis is to contribute important knowledge on the end-users' side and bridge the gap between system or security-oriented privacy research and the user-oriented privacy preferences modeling. Within the context of mobile app privacy, we are aiming to solve two key research questions that potentially can also be applied to other relevant domains. The first one is how can we convey mobile apps'

[1] Previously called "the Android Market."

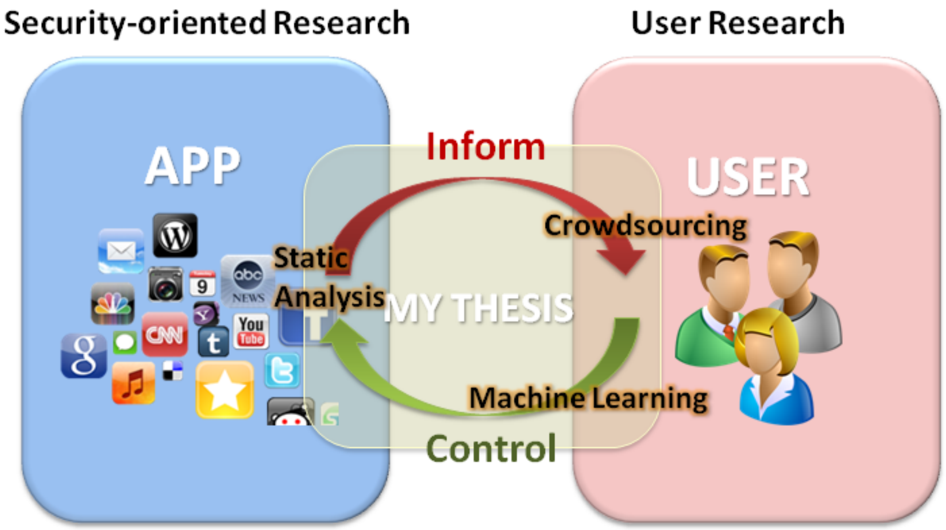


Figure 1. The fundamental goal of this thesis is to bridge the gap between security-oriented research and user research, emphasizing how to properly inform users of apps' privacy-related behaviors and how to help users control their privacy settings without burdening them with numerous decisions.

privacy-related behaviors to users in a more effective and understandable way. The other key research question is whether it is possible to simplify decisions users have to make without reducing their level of control over the decisions they really care about. In other words, this thesis focuses on two types of relationship between apps and users as shown in Figure 1, i.e. what and how should apps (or app markets) inform users regarding their data collection and usage practices (the "notify" arrow in Figure 1) as well as how to assist users in configuring their privacy settings to control the data usages of various apps (the "control" arrow in Figure 1).

Specifically, this thesis involves the detailed analysis of over 100,000 mobile apps and a collection of more than 20,000 responses from over 700 hundred smartphone users. We leverage static analysis to identify the 3rd-party libraries that bundled with apps to infer the use of sensitive data[2], crowdsourcing to collect users' privacy preferences at large scale, and machine learning techniques to isolate distinguishing patterns within apps' behaviors, as well as users' preferences. In these ways, we explore whether we can identify the key information to inform users and whether it is possible to reduce and simplify the number of privacy decisions exposed to users without negatively impacting their sense of control. The central thesis aims at providing quantitative foundations and user perspectives to mobile privacy research, which can be summarized as:

> ***By combining static analysis, crowdsourcing and user-oriented machine learning techniques, we can build accurate and understandable models of mobile app permissions and of users' willingness to grant these permissions.***

[2] Since the uses of 3rd-party libraries to some extent indicate why sensitive resources are used and the parties who collect this information.

> ***These models can in turn inform the design of more usable mobile app permission interfaces.***

Given the scale of data we are aiming to deal with and the goal of eventually applying our models to the real settings, i.e. to a market of more than one million of apps and hundreds of millions of users, scalability is an important criterion in the design and conduct of our studies and analyses. In this thesis, we will also show how we resolve this challenge by leveraging the power of cloud, crowdsourcing and unsupervised learning.

In the next section, we give a brief introduction to each of the three major components in this thesis outlining the key techniques we used and the lessons we learned in each step.

1.2 Three Major Components of this Thesis

Based on the objectives and the techniques involved, this thesis can be naturally divided into three components. In the first component, we describe the techniques we used to dissect and analyze mobile apps at a relatively large scale in order to understand the typical patterns of how apps consume users' sensitive personal data. In the second component, we describe our accomplishment in improving the privacy notification interfaces to convey richer and more pertinent information to users. In the last component of this thesis, we present our exploration in quantitatively modeling users' privacy preferences to identify representative privacy profiles that could greatly simplify the privacy configuration process.

1.2.1 Analyzing Apps' Privacy-Related Behavior

In Chapter 3 and 4, we describe the detailed procedures involved in downloading and analysis over 100K mobile apps. Specifically, we discuss how we use Androguard [2] – an Android reverse engineering tool to perform static code analysis on apps, focusing on identifying the sensitive data requested as well as the 3rd-party libraries that bundled within apps that consume these sensitive resources. We leverage the Amazon EC2 cloud to enable the batch processing to speed up the analysis of this large quantity of apps. To identify the purpose for which access to sensitive user data or phone functionality is requested, we identified the 400 3rd-party libraries that are most frequently used in all these apps and organized them into 9 categories. These categories include Targeted Advertising, Customized UI Components, Content Host, Game Engine, Social Network Sites (SNS), Mobile Analytics, Secondary Market, Payment and other Utilities. We also analyze how different types of resources (permissions) are used for various purposes.

We further performed clustering analysis to identify clusters of apps that request similar combinations of permissions. Our analysis identifies five different categories of apps, each exhibiting distinct patterns of permissions and purposes associated with these permissions. These different app categories give rise to different privacy risks and, as such, can also be expected to also give rise to different privacy preferences among users.

This component provides a systems-oriented foundation for us to better understand mobile apps in terms of their privacy-related behaviors, which enables us to study users' preferences in regard to these app behaviors in the later part of the thesis.

1.2.2 Improving Ways of Notifying Users

Previous studies have found that Android's existing permission interfaces are not sufficient for users to make informed decisions [54, 77]. In Chapter 5, we discuss how we identify essential information that needs to be conveyed to users, how we obtain this information, as well as how to present this information in more appropriate layouts.

More specifically, we frame mobile privacy in the form of people's ***expectations*** about what an app does and does not do as a key feature to convey to users, focusing on where an app breaks people's expectations. The other key feature we found crucial is the **purpose,** i.e. why the sensitive data is required, since people's perception of whether an app's permission is reasonable is strongly influenced by the purpose associated with this permission (e.g. internal use of one's location versus sharing that location with an advertising network). We show, how using crowdsourcing it is possible to collect this information and develop deep user privacy preference models that capture not just a user's willingness to grant a permission to an app but also the purpose associated with this permission.

Furthermore, based on our crowdsourced data, we present the design and evaluation of several new privacy notification interfaces that highlight the two key features we identified, including one preliminary design that adopted a similar text-based style as the existing Android permission screen and three other interfaces that visualize this information in more compact and understandable layouts.

1.2.3 Helping Users with Privacy Settings by Providing Privacy Profiles

In Chapter 6, we provide comprehensive quantitative modeling of users' privacy preferences. We extend our crowdsourcing study to a sample of over 1200 app-permission-purpose triples identified using static analysis. We collect over 20,000 subjective responses of these sensitive data usages from over 700 participants as our dataset to analyze users' privacy preferences. By performing clustering analysis, we show that it is possible to accurately capture the preferences of these users by subdividing them into four different groups of like-minded users. Looking at the different preference profiles associated with these groups, as identified by their willingness to grant different app-purpose-permission triples, we respectively label them the *conservatives*, the *unconcerned*, the *fence-sitters* and the *advanced users*.

We proceed to show that using the resulting four privacy profiles and simple decision trees to identify which profile best matches each user, it is possible to predict a user's willingness to grant app-purpose-permission triples with a high level of accuracy. This in turn offers the prospect of empowering users to better control their mobile app permissions without requiring them to tediously review each and every app-purpose-permission- decision associated with the apps they download on their smartphones, opening the door to privacy interfaces that could one day help reconcile privacy and user burden.

1.3 Research Contributions and Future Prospects

In short, this thesis contributes to mobile app privacy research in several ways including:

- Through a static analysis of over 100,000 apps that identified the 3rd-party libraries bundled in these apps, we contributed a valuable dataset to the community that includes attributes describing privacy-related behaviors of mobile apps, highlighting the purpose why users' sensitive resources are used.
- By clustering analysis of apps' privacy-related behaviors, we provided a new way to classify mobile apps based on how and why they use users' sensitive resources.
- We identified two key features --- expectation and purpose--- that greatly impact users' privacy preferences and should be conveyed to users for making better privacy decisions.
- We demonstrated the feasibility of using crowdsourcing as a compelling technique to examine people's preferences efficiently.
- We proposed a set of privacy interfaces that provide detailed explanations of apps' privacy-related behavior and leverage the misconceptions about an app that identified by crowdsourcing.
- We identified four groups of users with distinct privacy preferences of mobile apps' privacy-related behaviors.
- We generated a set of default privacy settings based on identified user clusters and demonstrated the potentials of these privacy profiles in terms of estimating users' privacy preferences more accurately and the great reduction of user burden they lead to.

Collectively, these contributions should provide a scientific basis for starting to reconcile mobile privacy and usability and, in particular, helping inform the design of more usable privacy interfaces and settings. At the end of this thesis, we also outline several directions that worth exploring in the future. These include leveraging NLP or other techniques to generate more functionality-related attributes for app analysis, a series of user studies to evaluate identified privacy profiles as well as the design and implementation of a privacy wizard that can bootstrap users' privacy settings.

In the next section, I will present a comprehensive summarization of literature that is related to smartphone privacy, as well as other relevant domains.

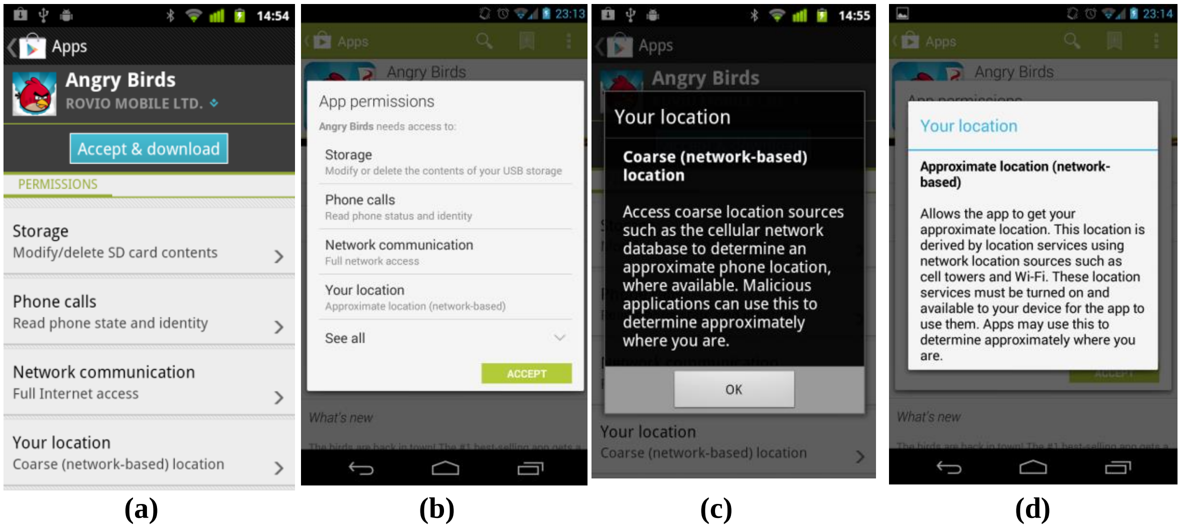
2 BACKGROUND AND RELATED WORK

In this section, I will first summarize the existing privacy frameworks of the two major smartphone operating systems, Android and iOS, to illustrate the problem and challenges that motivate my thesis. Then, I will survey all the recent work in mobile app privacy both from a systems, security, and end-user point of view. Following this, I will summarize the user privacy preference modeling research in other application, such as location sharing. Finally, I will mention all the other related technologies that inspired this work.

2.1 Android & iOS Privacy Frameworks

The Android permission framework is intended to serve two purposes to protect users: (1) limit the access of mobile apps to sensitive resources and (2) assist users in making trust decisions before installing apps. The latest Android 4.3 platform defines 11 permission groups with more than 130 permissions [62]. Android apps can only access sensitive resources if they declare permissions in the manifest files and obtain approval from users at installation. At the official Google Play store, before installing an app, users are shown a permission screen that lists resources an app will access. It is this information that users must use to decide whether to trust the app (see Figure 2). In order to proceed to installation, users need to accept all the permissions. Once granted, permissions cannot be revoked unless the user uninstalls the app.

Although intended to be a more open platform, Android's privacy framework puts the responsibility on users to make the "right" decisions. Therefore, its design easily suffers from two problems. One is the usability issue: several studies have pointed out that Android users generally paid limited attention to permission screens and had poor understanding of what the permissions implied [54, 77, 122]. Although Google has been continuously improving the ordering of permission groups and permission description,



Storage
Modify/delete SD card contents

Phone calls
Read phone state and identity

Network communication
Full Internet access

Your location
Coarse (network-based) location

(a)

App permissions
Angry Birds needs access to:

Storage
Modify or delete the contents of your USB storage

Phone calls
Read phone status and identity

Network communication
Full network access

Your location
Approximate location (network-based)

See all

(b)

Your location

Coarse (network-based) location

Access coarse location sources such as the cellular network database to determine an approximate phone location, where available. Malicious applications can use this to determine approximately where you are.

(c)

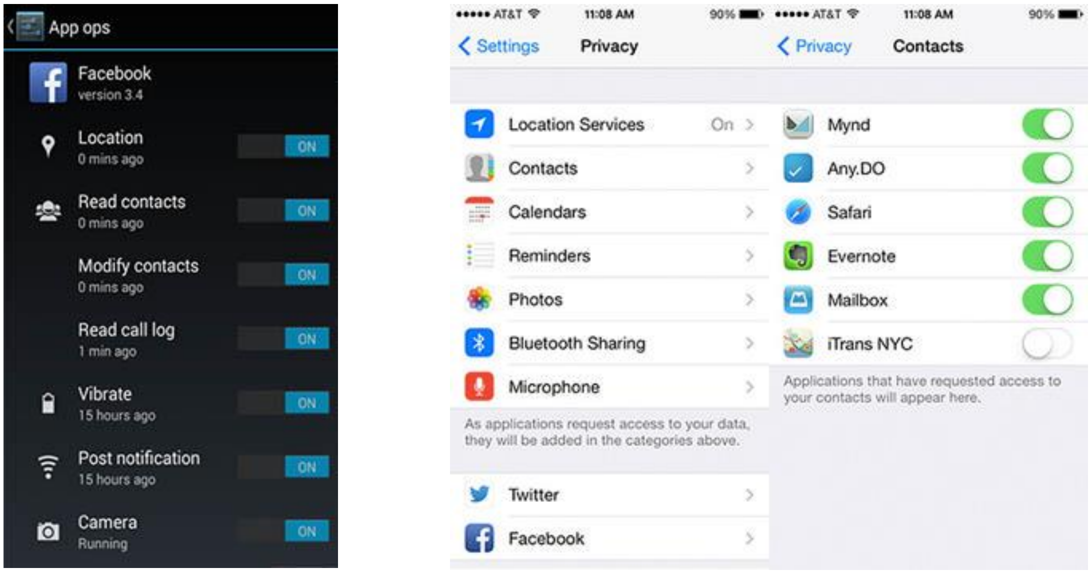
Your location

Approximate location (network-based)

Allows the app to get your approximate location. This location is derived by location services using network location sources such as cell towers and Wi-Fi. These location services must be turned on and available to your device for the app to use them. Apps may use this to determine approximately where you are.

(d)

Figure 2: The latest two generations of permission screens in Google Play Store (a) and (b). When a user clicks on an entry from the permission list, more explanations are shown (c) and (d). Previous research showed that most users click through the



App ops

Facebook
version 3.4

Location
0 mins ago
ON

Read contacts
0 mins ago
ON

Modify contacts
0 mins ago
ON

Read call log
1 min ago
ON

Vibrate
15 hours ago
ON

Post notification
15 hours ago
ON

Camera
Running
ON

●●●●● AT&T 11:08 AM 90%

< Settings Privacy

Location Services On >

Contacts >

Calendars >

Reminders >

Photos >

Bluetooth Sharing >

Microphone >

As applications request access to your data, they will be added in the categories above.

Twitter >

Facebook >

●●●●● AT&T 11:08 AM 90%

< Privacy Contacts

Mynd ●

Any.DO ●

Safari ●

Evernote ●

Mailbox ●

iTrans NYC ○

Applications that have requested access to your contacts will appear here.

Figure 3. Both of the two major smartphone operating systems provide (start to provide) users with finer privacy control over sensitive resources. The left screen shows the hidden “App Ops” permission Controls in Android 4.3 and the right screenshot shows privacy settings in iOS 6.

the current permission screens still generally lack adequate explanation and definitions. The other problem is the lack of controls. Before Android 4.3, once permissions are granted, users have no control over the permission usage of individual app other than uninstalling the app. This frequently puts end-users in a dilemma of trading off their privacy for functionality or cost, such that accumulatively users might lose confidence in Android system. The good news is, according to latest reports [20, 114], in Android 4.3 users are finally able to fine-tune their privacy preferences after installing apps by using a hidden “App Ops” feature (Figure 3, left). On the other hand, given the number of apps installed on an average users’ phone[3] and the number of permissions requested by each app, configuring permissions one by one seems infeasible, such that educating users to make proper configurations and developing trusted tools to provide certain level of automation become more and more important.

The privacy framework in iOS adopts a different strategy. Apple’s App Store does not present any data usage related information to users at install time. Instead, users are prompted to accept or deny the use of sensitive resource the first time it is used. In the latest iOS versions (iOS 5 and above), users also have the ability to turn on and off the data usage (such as location, contacts, calendars, photos, etc, see Figure 3, right) for each individual app, which is similar to what “App Ops” provides for Android system, and hence suffers from similar potential usability problems as well. For jailbroken iPhones, Protect My Privacy (PMP) provides users with even more controls [13, 19]. Whenever sensitive resources are requested, an alert is shown to the user with “protect” and “allow” option. Instead of merely blocking access to information that might cause unexpected behavior or crash, PMP supplies fake replacement data. In the privacy setting page, PMP

[3] Nielsen reported that US smartphones had an average of 41 apps installed in 2012 [87].

also provides an automated way of making privacy decisions by crowd-sourcing recommendations from other users. In my thesis, I will demonstrate that the average recommendations are not optimal for individual users since users' privacy preferences can differ from each other significantly.

In short, this thesis is motivated by these above-mentioned problems in these most popular smartphone operating systems and aims to leverage app analysis, crowdsourcing and user preference modeling to provide practical solutions to these problems.

2.2 Security-oriented Approaches in Mobile App Privacy

To protect users' privacy, a lot of work leveraged software security technologies to approach this problem, resulting in a number of useful tools for researchers and analysts to obtain deeper understanding of these mobile apps' sensitive behaviors.

To handle the increasing rate of malware in the Android market, in Feb 2012, Google announced their "Bouncer" service that scans apps for malware, spyware, Trojans, and other suspicious behaviors [124]. Though there is very little published information about how "Bouncer" actually works and how effective it is, the experiments conducted by Oberheide [96] unveiled that "Bouncer" performs dynamic runtime analysis of Android apps in an emulated Android environment and can be easily bypassed. For what we care, "Bouncer" was intended to detect malicious apps rather than privacy intrusive apps.

Researchers have also developed many useful techniques and tools to detect sensitive information leakage in mobile apps [18, 24, 29, 36, 45-49, 53, 55, 68, 112, 115, 123]. Three methods are usually used in app analysis, namely permission analysis, static code analysis, and dynamic flow analysis. Table 1 categorizes previous research and studies based on methods used and highlights the pros and cons of each method. There have also been a good number of security and privacy extensions proposed in recent years which aimed to give users more controls over sensitive resources on their smartphones [19, 29, 73, 93, 99, 123]. We will discuss these pieces of related work in more detail below.

2.2.1 Permission analysis

By analyzing the permission lists declared by app developers, potentially risky functionalities can be identified. This line of research has focused on how different permissions are used [24, 49, 55, 115] and highlights common usage patterns [24], misuses [53, 115], and potential implications to Android security and privacy [49, 53, 55]. Enck et al. [48] were the first to conduct permission analysis on the Android system. Among the 311 apps they examined, 10 apps were flagged with questionable private resource usage. Barrera et al. [24] performed permission analysis of 1,100 free applications in the Android Market and identified the exponential decay distribution in the number of applications that requested individual permissions (i.e., most applications require only a small number of permissions). Felt et al. [53] studied the effectiveness of Android's install-time permission. Specifically, they found that developers sometimes made mistakes in declaring permissions requests (e.g., requesting unnecessary permission, non-existing permission, etc.). Hence, in follow-up work [49], Felt et al. proposed the Stowaway tool, which performs static analysis to detect over-privileged

	Permission Analysis	**Static Analysis**	**Dynamic Analysis**
Examples	Enck’09 [48] Barrera’10 [24] Felt &Greenwood’11 [53] Felt&Chin’11 [49] Vidas’11 [115] Book’13 [31] Frank’12 [56]	Egele’11 [43] Chin’11 [36] Felt&Wang’11 [55] Enck’11 [47, 54] App Profiles [18]	Thurm’11 [43, 112] Enck’10(TaintDroid) [46] Beresford’11 [29] Zhou’11 [123] Hornyack’11 [68] Yang’12 [122]
Pros	Simple and efficient	Easy to automate, cover all possible execution patterns	Capture what actually happened, easy to interpret
Cons	Only high-level analysis cannot tell the whole story	Depend on decompiler, “Dead code” problem, i.e. segment of code never execute in the runtime;	Require human intervention, hard to automate

Table 1 : Categorization of existing work in mobile app analysis based on methodologies. The pros and cons of each method are highlighted. All methods assessed mobile apps’ behaviors from traditional security perspectives that cannot infer users’ perceptions of mobile privacy. Our proposed work makes use of the app analysis tool to obtain ground truth of mobile apps, aiming at bridging the gap between app analysis and users’ privacy preferences learning.

applications. Similarly, an Android SDK extension was developed by Vidas et al. [115], which assisted Android developers in including the minimum set of permissions required by their app’s functionality.

In more recent work, Frank et al. presented their results in mining permission request patterns of Android app in [56]. They identified over 30 typical patterns of permission request by using matrix factorization techniques. Our work differs from theirs in the sense that we enrich the dataset by including features describing why these permissions are requested. The work by Book et al. focused on how mobile behavioral advertising libraries use permissions over time by surveying 144K mobile apps [31]. They found that the ad libraries’ use of permissions has significantly increased over the last several years. Their excessive use of sensitive data poses particular risks to user privacy and security. In short, permissions are valuable for performance efficient security analysis; however, permission lists could not provide detailed information concerning what purpose private resources would be used, hence could only capture limited security and privacy risks.

2.2.2 Static analysis

Static program analysis can be conducted with or without source code. To date, most mobile app static analyses rely on decompilers to recover source codes of apps (e.g., [17, 97]). Egele et al. [43] proposed PiOS to perform static taint analysis on iOS application binaries to identify potential privacy violations. Among the 1,400 apps studied, more than half leaked the privacy sensitive device ID without the users' knowledge. Chin et al. [36]

proposed ComDroid, which operates on used disassembled DEX bytecode. Specifically, ComDroid identifies vulnerabilities in intent communications between applications, such as broadcast theft, service hijacking, malicious service launch, etc. Among 100 apps analyzed, Chin et al. found 34 exploitable vulnerabilities. App Profiles [18] developed by the RobustNet research group at the University of Michigan analyzed mobile applications offline to detect privacy-related actions written into the application source code.

While static analysis provides a complete and automated scan of mobile apps, its accuracy might highly depend on the performance of the decompiler used or the coding style of the developer. In addition, static analysis might produce false positive or false negative if the decompiled source codes contain what we referred as “dead code” (i.e. segment of program never executed in the runtime). Another challenge for privacy research involving static analysis is that this method cannot automatically determine whether privacy-related behavior is reasonable or not from users’ point of view.

2.2.3 Dynamic analysis

Dynamic analysis can help resolve ambiguity in permission granularity as well as provide an intuitive way to monitor how applications run. The *Wall Street Journal* reported the results of 101 popular smartphone apps for iPhone and Android devices that were examined by monitoring network analyses [112]. Results showed that 56 apps transmitted the phone's unique ID to third party servers without user consent, and 47 apps transmitted the phone's location and other personal information such as age, gender, etc. TaintDroid [46] performed a thorough dynamic flow analysis to capture information leakage on Android devices in real time. The authors modified the Android's Dalvik VM to perform instruction-level taint tracking that captures how private information flows from its source to its destination (i.e., network interface). Other work has built on TaintDroid to provide more pertinent privacy analyses or controls [29, 68]. The work by Yang et al. integrate crowdsourcing into dynamic analysis to understand why certain permissions are required [122]. They paid crowd workers to compare the screenshots of apps with and without granting permissions and summarize the differences in order to identify the purpose of accessing sensitive data such as for serving ads or for providing context-aware services.

Dynamic analysis identifies what actually happens when an application is running. One drawback of dynamic analysis is that it is limited by scalability because human interventions (interactions with mobile apps) are needed to trigger certain behaviors of the apps in the process of analysis.

Though app analysis provides us with a better understanding of apps’ behaviors, it cannot infer people’s perceptions of privacy or distinguish between behaviors which are necessary for an app’s functionality versus behaviors which are privacy-intrusive. Our work complements this past work by suggesting an alternative way of looking at mobile privacy from the users’ perspective by leveraging crowdsourcing to bridge the gap between app analysis and resolving users’ privacy concerns. To achieve this goal, we opt to use static analysis to capture the ground truth of apps with regard to type and purpose of information disclosed because of the scalability issue.

2.2.4 Security and privacy extensions

All these approaches provide useful means to dissect mobile apps providing more and more detailed information on how they consume users' sensitive information, the results of which also outlines the potential privacy and security risks of specific usage patterns. Upon these findings, many security extensions have been developed to harden privacy and security of smartphone operating systems.

MockDroid [29] and TISSA [123] substituted fake information into API calls made by apps, such that apps could still function, but with zero disclosure of users' private information. Similarly, ProtectMyPrivacy [19] on jailbroken iPhone also enable users to substitute fake information to protect their privacy. In addition to faking information, AppFence [68], a subsequent project of TaintDroid, allowed users to specify which resources should only be used locally. It also hashed the phone identifiers in a way that it no longer could be linked to users, while still being useful for application developers to track application usage. Nauman et al. [93] proposed Apex, which provides fine-grained control over resource usage based on context and runtime constraints such as the location of the device or the number of times a resource has been used. They implemented an extended package installer named Poly that allows users to specify their policy at time of installation.

To enable wide deployment, Jeon et al. proposed an alternative solution that rewrote the bytecode of mobile apps instead of modifying the Android system [73]. When accessing sensitive resources, the modified apps talk to a privacy proxy layer instead of directly talking to Android APIs. Pearce et al. [99] proposed to adopt privilege separation for mobile applications and advertisers in Android OS, which is motivated by the fact that over 56% of apps uses users' location information only for serving ads. They suggested unifying all the mobile ad libraries into a system service that can be integrated into the Android platform. In their proposed AdDroid framework, a new permission ADVERTISING needs to be declared by app developers when a mobile app wants to deliver ads to users. Although the techniques they proposed are sound and effective, given the existing mobile app ecosystem, advertising companies have little incentive to cooperate in this initiative.

These proposed privacy extensions aimed to provide users more control over apps and assumed that users are able to configure these settings perfectly. However, this assumption was not grounded by user studies. Dumping these settings on users and relying on users to specify their privacy preferences without adequate information could be questionable or even counterproductive.

2.3 End-User Research in Mobile App Privacy

In contrast to the above systems-oriented approaches, another important facet of privacy research approaches the challenge from the end-users' side. In this line of work, researchers tried to gain deeper understanding of users, including their biggest privacy concerns, their perception of mobile apps, as well as their preferences of different types of sensitive data usages.

Several user studies have examined usability issues of permission interface displayed to users before downloading apps. Kelley et al. conducted semi-structured interviews with Android users and found that users paid limited attention to permission screens and had a poor understanding of what the permissions implied [77]. Specifically, permission screens generally lack adequate explanation and definitions. Felt et al. [54] found similar results from Internet surveys and lab studies that current Android permission warnings do not help most users make correct security decisions. In later work, Felt et al. [51] surveyed more than three thousand smartphone users about 99 risks associated with 54 permissions without considering specific apps. Their survey focused more on security risks that malicious apps can exploit rather than the potential privacy concerns caused by normal mobile apps.

An interview study by Chin et al. [37] probed smartphone users' concerns and fears with regard to privacy and security and offered several recommendations that could mitigate these threats. They found that users are in general more concerned about their privacy on their smartphones than their laptops in performing tasks such as payment and online banking etc. The work done by Jung et al. [22, 74] included lab studies and qualitative interviews to evaluate the gaps between user expectations with respect to mobile app privacy. They found users were surprised by the amount and frequency of data leaving their phones. There were three types of unanticipated data use, including discreetly collecting personal data in the background; application collecting seemingly unnecessary data with respect to their functionality; application collect excessive amount of personal data (frequency). Egelman et al. performed experiments to gauge how smartphone users value their privacy [44]. They found that 25% users are willing to pay a $1.50 USD for the application requesting the least permissions. Around 80% of participants stated that they would be willing to receive targeted advertisements regardless of the permissions used if it would save them $0.99. Benenson et al. surveyed over 700 German students to compare users’ security and privacy perceptions of Android and iOS [25]. Their data suggested that (1) if users are brand-aware, then they are more likely to have an iPhone; (2) Having an Android phone is positively correlated to being more privacy aware; (3) Female users are more likely to have an iPhone.

Methodology-wise, Felt et al. discussed the strengths and weaknesses of several permission-granting mechanisms and provided guidelines for using each mechanism [50]. They suggested that for different types of sensitive data, different permission-granting mechanisms should be independently triggered and the permission-granting process should try to avoid interrupting user’s primary tasks.

With regard to privacy interfaces, Kelly et al. proposed to improve Android’s existing permission screen by putting the privacy facts inline with the app’s description [78]. They also suggested including how the app used several types of personal information, including contacts, location, calendars, credit cards, diet, health, photos etc. They demonstrated that users who saw the new design were more likely to pick the application that requested fewer permissions than who saw the existing Android permission screen. Choe et al. contributed to the privacy interface design by investigating whether framing effect can be used to nudge people away from privacy invasive apps [38]. They found that between semantically equivalent visuals, different framing methods (positive framing and negative framing) did not affect the time users spent on privacy interfaces; however, a positive framing icons were more effective in making a low privacy rating app look more unfavorable, whereas negative framing icons were more effective in making a high privacy rating app more unfavorable.

The National Telecommunications and Information Administration (NTIA), the agency of US department of Commerce that serves as the President’s principal adviser on telecommunications policies, released guidelines for a short-form privacy notice as a voluntary Code of Conduct in July 2013, aiming to provide app users with an easy to understand display indicating which categories of personal data may be collected by the app and which types of entities those data may be shared with [67, 95]. The Code of Conduct identifies 8 categories of personal data categories, including for example, Internet browsing history, phone and text logs, contacts, financial information, location, and more. It also identifies 8 types of entities with whom personal data might be shared, including ad networks, data analytics companies, government entities, social networks, and more. Note that the Code of Conduct provides several general design guidelines and required explanatory text, but does not specify a particular standardized design at this point. Past work has looked at the usefulness and understandability of the category names used by NTIA [23]. Based on the code of conduct, several notice screen mockups have been proposed, such as [67]. In the collaborative work with Wong et al. [120], we evaluated one of the NTIA mockup by testing participants’ understanding of examples of this interface. We found that this interface is not as understandable as expected. It suggests that if NTIA’s guidelines are adopted, much more work needs to be done to improve the visual displays.

All of the above-mentioned work provided valuable insights into users’ privacy concerns. This thesis provides a more quantitative approach, by leveraging the power of crowdsourcing, we built a dataset contributed by over 700 participants with their opinions over 1200 app- permission-purpose triples to uncover the underlying patterns of users’ privacy concerns. This dataset enables an in-depth probing of users’ mobile app privacy preferences.

2.4 User Modeling in Location Sharing

Our initial exploration of users’ mobile privacy preferences started with location sharing, focusing on understanding and resolving users’ privacy concerns when using location sharing applications (LSAs). These types of applications facilitate and encourage users to share their location information with others. They have recently attracted interest from both industry and academia [5, 8-12, 16, 32, 60, 69, 70, 98, 107, 116, 119]. With the proliferation of smartphone ownership, most location-sharing services are available on mobile platforms (e.g., Google Latitude [10], Foursquare [9], Facebook Places [8]). As a special subset of mobile apps, where the users’ location information is primarily consumed by people in their social networks,[4] studying the privacy issues in LSAs could provide important lessons from both methodological perspective and knowledge perspective.

[4] Though some location-sharing mobile apps also transmit users' location information to ad networks for advertising purposes.

Some of my past work fall into this line of research [83, 84, 110]. Our findings indicated that even only considering one type of sensitive resource users' privacy preferences could be very complex and were influenced by different factors [27, 107]. For example, by tracking 26 participants for 3 weeks and asking them to provide place names for each location for various sharing scenarios, we observed that people modulate the way they convey location information to cope with privacy concerns [84]. They generally used two major techniques to tailor their location information, i.e. describing it semantically or geographically. Multiple factors were considered when users decided on what to disclose, including their relationship with the recipients, their perceived levels of comfort in sharing specific locations, the recipients' familiarity with the places, and place entropy.[5]

Along a similar direction, in collaborative work with Tang [110], we compared the location names users selected in different scenarios and reframed the location-sharing applications (LSA) into two categories, based on the users' intention of sharing, namely purpose-driven LSAs and social-driven LSAs. Our findings indicated that people have distinct sharing preferences given the purpose of sharing (1) the types of location information they chose to share, (2) the different privacy concerns people had and strategies used to cope with these concerns, and (3) how privacy-preserving these location disclosures were. In the problem of mobile app privacy, the purpose of information disclosure remains an important factor that influencing people's decision.

In other work, we looked at the effect of cultural differences in location sharing. In [83], we reported findings of a three-week comparative study collecting location traces and location-sharing preferences from two groups of university students in the U.S. and China with similar demographics. We found that, on average, Chinese participants were more conservative about sharing their location; however, when they were given the ability to control the granularity of sharing, they shared more detailed location compared to U.S. participants. This finding suggests that, in the absence of granularity settings, U.S. participants were more willing than Chinese participants to relax their preferences and share their finest location details even when doing so was not their optimal choice, whereas Chinese participants were more likely to do the opposite. A significant implication of this finding is that granularity settings are likely to be more important for the adoption of location sharing among Chinese users than among American users.

There is also a line of work focused on a more quantitative approach to modeling users' location sharing preferences. For example, in the above-mentioned work [84], we also demonstrated the feasibility of applying machine learning techniques to predict the way people manipulate the disclosure of their location information in different context (e.g. based on how far away they are). This work suggested that people's privacy preferences though complicated can still be modeled quantitatively. The work by Ravichandran et al. [103] learnt a set of default policies from users' location sharing preferences using decision-tree and clustering algorithms. They suggested that providing users with a small number of canonical default policies to choose from can help reduce user burden when it comes to customizing the rich privacy settings they seem to require. The work by Cranshaw et al. [39] used a classifier based on multivariate Gaussian mixtures to

[5] Place entropy characterizes the diversity of users seen in a particular place. See [40]

incrementally learn users’ location sharing privacy preferences. Kelley et al [79]and later Mugan et al. introduced the notion of understandable learning into privacy research [92]. They used two types of user-oriented machine learning techniques, namely default personas and incremental suggestions, to identify users’ privacy rules, resulting in a significant reduction of user burden. By restricting the level of control the user has over the policy model, their algorithm produced accurate and understandable learning results. Wilson et al. [119] evaluated the impact of privacy profiles in a location sharing study. They observe that although participants were given the ability to refine their preferences, the impact of the initial privacy setting remained visible after several weeks of use. In addition, participants who were exposed to the privacy profiles were more inclined to share than those who were not.

Previous research (including my own work) has provided important knowledge in understanding users’ privacy concerns and needs in mobile context-sharing. Considering methodology, multiple user studies [26, 84, 113] have shown that remote auditing-based study methods (i.e., participants provide their responses remotely through a web site) is an efficient way to conduct privacy related studies. However, we are fully aware of the limited scalability of this approach given the number of mobile apps we want to investigate. Therefore, to tackle this challenge, we propose that mobile privacy user studies can take advantage of crowdsourcing to harvest users’ privacy preferences. We also learned that users’ location privacy preferences are dynamic and complex, but for the most part predictable. We demonstrate in Chapter 6 that this point also holds in the context of mobile app privacy. Furthermore, as pointed out by Wilson et al. in [119], “… the complexity and diversity of people’s privacy preferences creates a major tension between privacy and usability…”, and mobile app privacy poses similar usability challenges. In Chapter 6, we will demonstrate how we generate appropriate privacy profiles (default settings) for users as one possible way to simplify the decisions users have to make. As in Mugan et al. in [92], we also take into account understandability considerations in our work and aim to build understandable quantitative models of users’ mobile app privacy preferences. This is done using interpretability and generalizability as two criteria in our work on modeling users’ preferences.

2.5 Crowdsourcing and Human Computation

Crowdsourcing and human computation have gained attention as both a topic of and tool for research. Several methodological papers have addressed how to more effectively utilize crowdsourcing to yield better results [41, 71, 88, 90, 106]. Amazon’s Mechanical Turk (AMT)[1] is currently the most popular crowdsourcing platform and the one used in this work. With AMT, requesters can publish Human Intelligence Tasks (HITs) for workers. A number of projects have successfully used AMT and have ranged from human assisted online tasks (such as image labeling) to surveys and user studies [30, 57, 66, 85, 86, 121]. My thesis makes use of many of the findings and methodologies mentioned above and builds on past work by extending the use of crowdsourcing to a mobile privacy study. In doing so, we demonstrate the feasibility and potentials of crowdsourcing as a scalable tool for privacy studies.

2.6 Relationship to Prior Work

Before moving on to the details of this thesis, I want to point out a few distinctions between my thesis and past related work.

From a technology standpoint, this thesis does not aim to produce new tools. Instead, it demonstrates that by identifying third party libraries that most commonly found in mobile apps, it is possible to extend static analysis to identify the purpose associated with many mobile app permissions in a scalable manner. In addition, this thesis also links users' subjective feedback to various private resource usage patterns as identified through app analyses.

Meanwhile, the security extensions mentioned above do provide users with more control over private data; however, these designs are not grounded in adequate user studies. Specifically, we foresee that these granular controls might overwhelm users with too many privacy decisions to make and might ultimately be unusable in practice. This potential usability issue also motivates my work in assisting users with privacy configurations by providing meaningful default settings.

From an HCI standpoint, this thesis probes much deeper in the users' privacy decision processes compared to previous permission usability studies [54, 77] or privacy surveys and interviews [37, 51]. By performing clustering, we isolate five classes of mobile apps and four different groups of users with distinct characteristics. Each cluster of users can be interpreted in the form of a privacy profile describing users' different level of concerns over different data usages. These findings provide important practical suggestions to inform the design of simpler, easier-to-use interfaces and privacy control mechanisms that matter to users.

3 DISSECTING AND UNDERSTANDING THE BEHAVIOR OF SMARTPHONE APPS

Before analyzing people’s privacy preferences of mobile apps, it is necessary to gain a deeper understanding of mobile apps with regard to their privacy-related behaviors as well as the implication of these behaviors. In this chapter, I will provide technical details of how we obtained metadata and binary files of apps from Google Play, and how we decompile and analyze these apps on a large scale.

3.1 Data Gathering

We collected meta-information for 171,493 Android apps and binary installation files (also known as apk files) for 108,246 free apps available on Google Play in July 2012.

Each Android app in Google Play has its own description page. However, there is no index of apps that is publicly available. To build our dataset, we used a Python-based webpage crawler to run a Breadth-First-Search starting from Google Play's home page, and downloaded all of the web pages containing app description information when we traversed Google Play. Once we got a description page, we parsed the HTML page to extract the app's metadata, including its name, category, number of downloads[6], average user rating score, rating distribution, price, and content rating.

Next, for each app we crawled, we downloaded its binary installation file through an open-source Google Play API [3]. Google imposes limits on the number and the frequency of app downloads. To work around this limit, we dynamically switched among 20 different Android accounts to prevent being permanently banned from Google Play. Using the same API, we also downloaded a total of 13,286,706 user reviews which was used in a side project [59]. Note that Google has strict restrictions on app purchase frequency and limits the number of apps that can be purchased with a single credit card. Because of these restrictions, the binary files downloaded in this thesis work are all free apps. However, we believe that our approach and the majority of our findings applies to paid apps as well. The entire apps’ metadata takes up about 500MB of storage space when stored in a MySQL database and all the binary files take approximately 300GB of storage space on a disk.

At the time of writing, the aforementioned API [3] no longer supports the current version of Google Play. Readers interested in conducting similar studies in the future should explore alternative APIs such as [81].

3.2 Dissecting Android Apps

While dynamic analysis can provide information on apps’ runtime behaviors, the requirements of this type of analysis exceeded resources at our disposal, given the large number of apps we wanted to study. Instead, we opted to use static analysis tools given that they are more efficient and easier to automate. After examining several Android reverse engineering tools [2, 17, 3, 6, 7], we chose Androguard [2] as our major static

[6] Google does not provide the absolute number of downloads. Instead, it discretizes this number into several ranges.

analysis instrument. Androguard is a tool written in Python to decompile Android apk files and to facilitate code analysis with well-documented APIs. More specifically, Androguard suited out needs for the following reasons:

- Androguard is available for Linux/OSX/Windows as it is Python-powered. This gives us the flexibility to deploy our analyzer on a number of different types of servers.
- Androguard provides an efficient de-compilation functionality that can de-compile Dalvik bytecodes to Java source code faster than other de-compilers. Given the scale of the app analysis we planned on conducting, efficiency is crucial for us.
- Androguard allows analysts to create customized static analysis scripts to examine app's specific behaviors. In our case, since we are particularly interested in apps' privacy-related behaviors, this was a significant advantage.
- Androguard allows batch processing of analysis tasks, which facilitates the automation of analysis tasks.

We created our own analysis script with the Androguard library and identified the following information related to apps' privacy-related behaviors.

- Permission used by each app.
- The destination and source classes involved in the use of permissions.
- All the third party libraries included in the app.
- Permissions required by each third party library.
- All the URLs and/or IP address the app is connecting to.

The permission usages tell us what type of sensitive user data apps are requesting. By analyzing the 3rd-party libraries in an app and what permissions these libraries use, we can infer if users' sensitive data is required for apps' functionality or for other purposes, such as for delivering targeted ads, for market analysis and for promoting sharing on Social Network Sites. The URLs help us to confirm the destinations where different user data sent to.

Permission information is directly obtained by parsing the manifest file of each apk. We further scan the entire de-compiled source code and look for specific Android API calls that request permissions to determine the destination and source classes involved in the use of these permissions.

Third party libraries are identified by looking up package structures in the de-compiled source code. From example, if we found a "com.flurry"[7] sub-folder inside the de-compiled source code, we say that the "flurry" library--- a mobile analytics library --- is included in this app. It is possible that we failed to identify some libraries, although we assume that we were able to correctly identify the most popular ones. We did not distinguish different versions of the same third party library to reduce the complexity of

[7] http://www.flurry.com/ Flurry analytics is a cross-platform analytics service for developers to understand how consumers interact with their mobile applications.

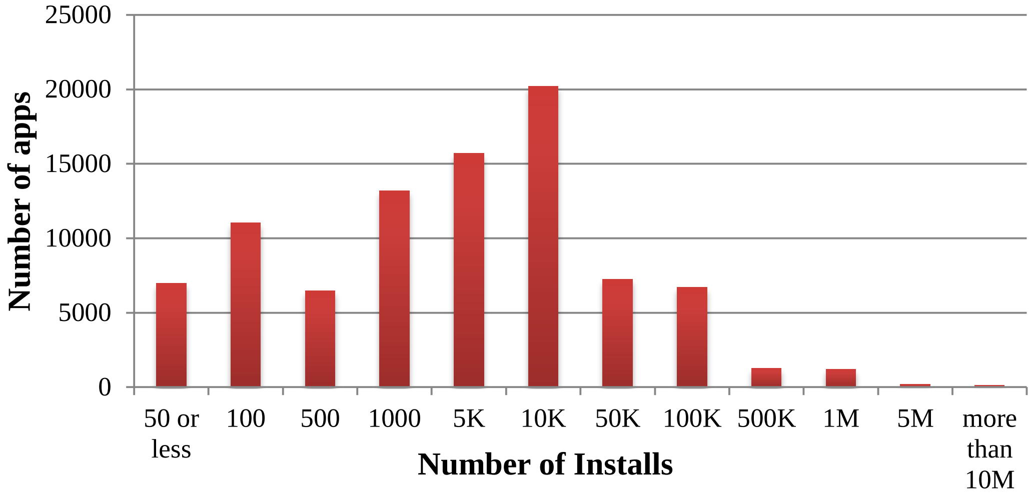


Figure 4: Distribution of apps in our dataset by number of installs.

our analysis. Similar to the permission analysis step described above, we determined the permission usages of each 3rd-party library by scanning through all the Android standard API calls that relate to the target permission in the de-compiled versions of the libraries' source code.

To scale up the analysis, we employed five Amazon EC2 M1 Standard Large Linux instances to perform batch processing of the static analysis. Each instance has the capacity of 4 ECUs[8] and 8 GB memory. The total analysis required 2035 instance hours, i.e. approximately 1.23 minutes per app. Among all the 108,246 free apps, 89,903 of them were successfully decompiled (83.05%). Upon manual inspection of a few failure examples, we were led to believe that failure to de-compile was primarily attributed to two reasons, (1) the binary files were corrupted during the download or transmission to cloud, (2) the binary files were intentionally obfuscated to prevent reverse engineering by using techniques such as APKProtect [4].

3.3 *Analysis Results*

Among the 89,903 free Android apps we decompiled, the percentages of each category are very similar to the stats reported in AppBrain [21]. Figure 4 shows the distribution of the apps according to the lower bounds on the total number of installs for each app. Google does not provide the absolute number of downloads; instead, it discretizes this number into several ranges. The x-axis of Figure 4 is labeled by the lower bounds of these ranges. Approximately 54.7% of apps had been downloaded more than 1000 times and less than 50,000 times. Since the data was collected in July 2012, the current number of downloads for each app might be much higher than the number plotted here.

Among the 89,903 free apps that were successfully analyzed, we identified over 500 different 3rd-party libraries used by various apps. We analyzed the top 400 most used 3rd-party libraries online to understand the purpose or functionality associated with each. We

[8] 1 ECU is the equivalent CPU capacity of a 1.0-1.2 GHz 2007 Opteron or 2007 Xeon processor.

Type	Count	Examples	Description
Utility	140	Xmlparser, hamcrest…	Utility java libraries, such as parser, sql connectors, etc
Targeted Ads	137	admob, adwhirl, greystripe…	Provided by mobile behavioral ads company to display in-app advertisements
Customized UI Components	29	Easymock, kankan, viewpagerindicator…	Customized Android UI components that can be inserted into apps.
Content Host	25	Youtube, Flickr…	Provided by content providers to deliver relevant image, video or audio content to mobile devices.
Game Engine	20	Badlogic, cocos2dx…	Game engines which provide software framework for developing mobile games.
SNS	15	Facebook,twitter, socialize…	SDKs/ APIs to enable sharing app related content on SNSs.
Mobile Analytics	14	Flurry, localytics..	Provided by analytics company to collect market analysis data for developers. (in recent years, mobile analytics libraries have also been used to deliver in-app ads)
Secondary Market	11	Gfan, ximad, getjar…	Libraries provided by other unofficial Android market to attract users.
Payment	9	Fortumo, paypal, zong…	e-payment libraries

Table 2: The types of 3rd-party libraries identified. Based on the types of services they provide, we categorize them into 9 basic categories.

eventually identified nine major categories of libraries as detailed in Table 2. Again, note that, we do not distinguish between different versions of the same library.

Among all the identified libraries, 34.5% of them are Java utility libraries, such as XMLparsers, SQL connectors, etc. Most of these utility libraries do not involve Android API calls. Accordingly these libraries do not require any Android permissions, though INTERNET permission is sometimes required to allow these libraries to communicate with external servers.

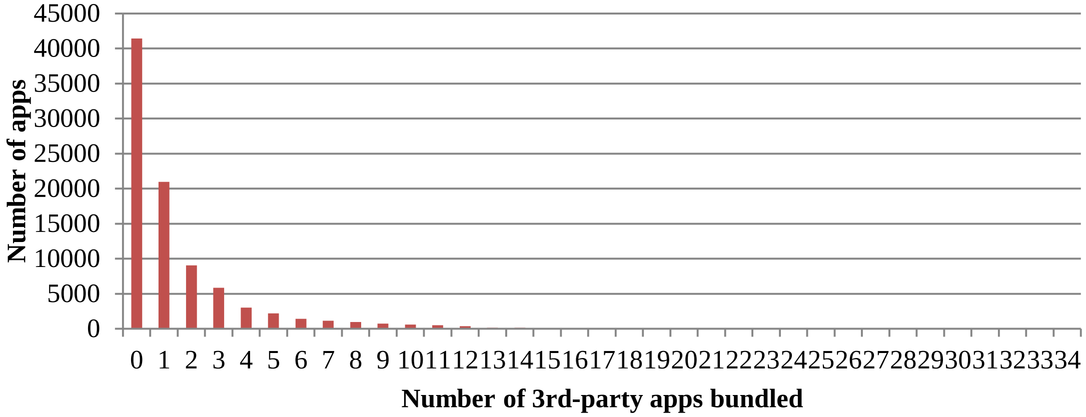


Figure 5: Distribution of total number of 3rd-party libraries an app bundled with. The x-axis shows the number of 3rd-party libraries, the y-axis the number of apps bundled with corresponding number of libraries. Majority of apps only have a small number of 3rd-party libraries bundled in.

The second largest category of 3rd-party libraries is targeted behavioral ads libraries, which are approximately one third of all the libraries identified[9]. Mobile targeted advertising is one of the major monetization channels for app developers, especially those who develop free apps. Java-based advertising libraries supplied by advertising agencies are bundled into application packages to deliver behavioral targeted ads based on users' interests.

These libraries communicate with servers controlled by advertising agencies, transmitting ad requests, displaying selected advertisements, and handling user interactions with these ads. In order to display ads that are more relevant to users, these targeted ads libraries usually have the ability to collect contextual information such as users' location, phone number, and other information that can imply users' preferences, which poses significant privacy concerns to mobile users [31].

The other seven categories include Customized UI Component libraries which are usually developed and published by 3rd-party companies or developers to promote the reuse of UI modules; Content Host libraries are usually supplied by companies who supply multimedia content online such as YouTube, Flickr, etc; Game Engine libraries are usually used by mobile game developers in their game design; SNS libraries are supplied by major Social Networking Sites to provide in-app sharing functionality. For example a music player app might allow the user to post information about the sound track the user likes to her Facebook wall through these type of libraries; Mobile Analytics libraries are provided by mobile analytics companies such as Flurry and Localytics, which gather and analyze the users' in-app interactions with the app on behalf of the app developers to identify who the customers are, where they come from and what they are doing; the remaining Secondary Market and Payment libraries are self-explanatory. Note that, after we crawled the dataset, a large number of new mobile analytic companies has emerged. In addition, many mobile analytics companies have started to integrate their services with

[9] The most used ad libraries we identified are similar to the ones reported by Book et al in [30]. Therefore, we do not repeat the stats here. For interested reader, please refer to their paper.

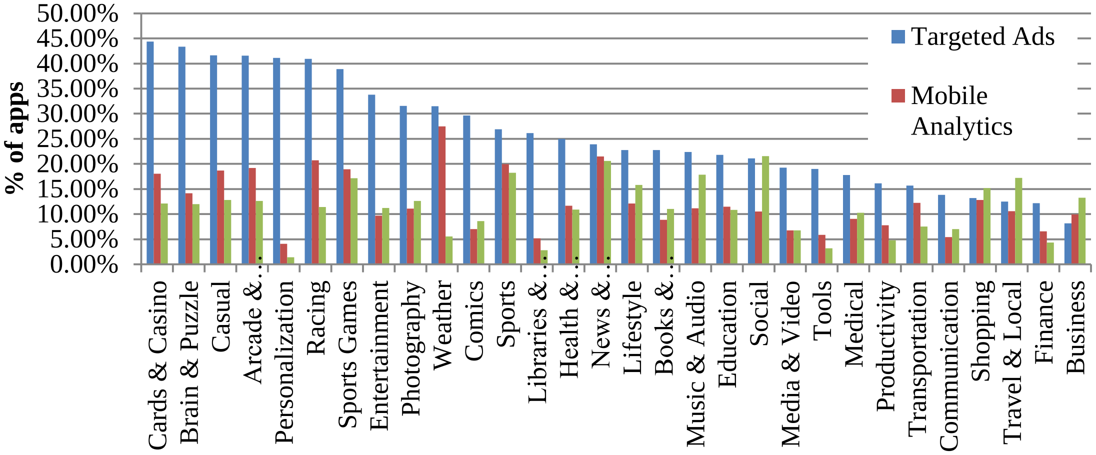


Figure 6: Penetrations of three types of 3rd-party libraries across 30 Android app categories. We see significant penetration of targeted advertising libraries (blue bars) in almost all categories. Mobile analytics and SNS libraries also have relatively high penetration.

mobile behavioral advertising. Therefore, though in this thesis we still distinguish Mobile Analytics and Targeted Ads libraries, we strongly recommend later work should combine these two categories together. As far as we can tell, no new categories has emerged at the time of writing this thesis.

For all the apps we analyzed, we observed an average usage of 1.59 (*SD=2.82, median=1*) 3rd-party libraries (See Figure 5) in each app. There are some extreme cases where an app uses more than 30 3rd-party APIs. For example, the app with the package name (com.wikilibs.fan_tatoo_design_for_women_2.apk) used 31 3rd-party libraries, 22 of them are targeted advertising libraries, such as *adwhirl*, *mdotm*, *millenialmedia*, *tapjoy*, etc. In the majority of the cases (91.7%), apps are bundled with less or equal to 5 different 3rd-party libraries.

We further breakdown the 3rd-party libraries across all the 30 app categories, emphasizing the penetration of three types of usage, namely the targeted ads, mobile analytics and SNSs. Figure 6 shows that for most popular categories such as mobile games, and personalization apps, the targeted advertising libraries are found in more than 40% of these apps, even the lowest ads-penetrated category--- Business apps--- there are more than 8% of them bundled with targeted advertisement libraries. SNS libraries closely follow up targeted ads libraries, achieved an average penetration of 11.2% of the app market. The Social category of apps has the maximum usage of this type of 3rd-party libraries, which makes sense. Mobile analytics libraries have an average penetration of 9.8% of the app market, and usually bundled with categories of apps that users use daily, such as weather, news & magazines, sports, etc.

Lastly, we report on the usage of several of the most sensitive permissions in terms of why they are required in apps (see Table 3). We focus our analysis on the top four major uses, which are:

- For internal use, where the permission triggering Android API calls are found within the application-specific code (rather than the bundled libraries). Given the

	Internal Use	Targeted Ads	Mobile Analytics	SNS
INTERNET	41.33%	47.48%	20.71%	16.30%
LOCATION	17.48%	72.94%	26.08%	6.07%
PHONE_STATE	24.55%	74.40%	16.04%	6.35%
READ_CONTACTS	52.07%	45.76%	-	2.81%
BLUETOOTH	86.54%	-	-	-
SMS	63.33%	38.81%	-	1.19%
GET_ACCOUNTS	32.51%	4.95%	-	8.04%
CAMERA	30.06%	17.45%	-	-
RECORD_AUDIO	91.91%	9.51%	-	-

Table 3: Distribution of permissions used for various purposes, including used for apps' functionality (internal use), for targeted advertising, for mobile analytics, and for social network sharing. E.g. 41.33% of apps that required INTERNET permission used this permission for internal use, 47.48% of them used for targeted ads. Note that, an app can use one permission for multiple purposes, and so rows do not sum to 100%.

limitations of our analysis, we cannot provide detailed explanations of why certain resources are used (such as "for navigation", "for setting up ringtone", etc.), since this level of detail currently requires a great deal of manual code inspection. In other words, with current technologies, we have no easy way to tell how developers actually use the data.

- For targeted ads, where the permission triggering Android API calls are found within targeted ads libraries. Relevant permissions might include INTERNET, ACCESS_FINE/COARSE_LOCATION, VIBRATION, and even CAMERA[10].
- For mobile analytics, where the permission triggering Android API calls are found within mobile analytics libraries. Usually, this type of libraries requires the access of INTERNET, ACCESS_FINE/COARSE_LOCATION and READ_PHONE_STATE. The last permission is used to obtain unique phone ID, as well as detecting if the user is on a phone call.
- For SNS, where the permission triggering Android API calls are initiated by the SNS libraries, such that users can share app relevant information together with other context information to social network sites.

Table 3 shows the distribution of permissions used for various purposes, such as for apps' functionality, for delivering targeted advertisements, for mobile analytics, or for sharing on SNSs. For example, we found that 41.33% of apps that required INTERNET permission used this permission for internal use, and 47.48% of them used for targeted ads. The numbers in each row do not necessarily add up to 100% since one permission can be used for multiple purposes in an app. We also notice that for permissions like ACCESS_FINE/COARSE_LOCATION and READ_PHONE_STATE, a significant portion of apps used these permissions purely for delivering targeted advertisements. In other words, a large portion of apps requested excessive permissions just for monetization purposes.

[10] We found this permission is required by one version of mobclix http://www.mobclix.com/ , a very popular mobile advertising library.

Cause	Attractiveness	Stability	Accuracy	Compatibility	Connectivity	Cost	Telephony	Picture	Media	Spam
Words	boring	closes	find	galaxy	log	free	unistall	pictures	video	ads
	bad	close	location	battery	error	money	want	picture	sound	notification
	stupid	load	search	support	account	buy	need	pics	watch	spam
	waste	every	info	off	connect	pay	send	camera	videos	bar
	dont	crashes	useless	droid	login	paid	messages	save	songs	notifications
	hard	keeps	data	nexus	connection	refund	delete	wallpaper	audio	adds
	make	won	way	compatible	sign	want	let	see	sounds	annoying
	way	start	list	install	let	back	contacts	photos	hear	many
	graphics	please	sync	samsung	slow	bounght	calls	upload	record	pop
	controls	closing	wrong	worked	website	waste	off	pic	anything	push
%	18%	13%	13%	11%	10%	9%	8%	8%	5%	5%
Example app	Stardunk	Opera	Kindle	App 2 SD	Zedge	Sygic	LINE	Pho.to	IMDB	Brightest Flashlight
	Blast Monkeys	Bible	Kobo	Solar Charger	Dropbox	Cut the Rope	WhatsAPP	Retro	Tuner	Shoot the Apple

Table 4: Most frequent words from the top 10 causes found by LDA topic modeling. The percentages in the middle row indicated the portions of apps that had comments expressed corresponding themes.

In short, static analysis with batch processing in the cloud enables us to dig deeper into apps' privacy-related behaviors and help us at some level understand better sensitive resource usages in terms of what are used and hints as to how and why they are used.

3.4 Other Potential Ways to Analyze Apps[11]

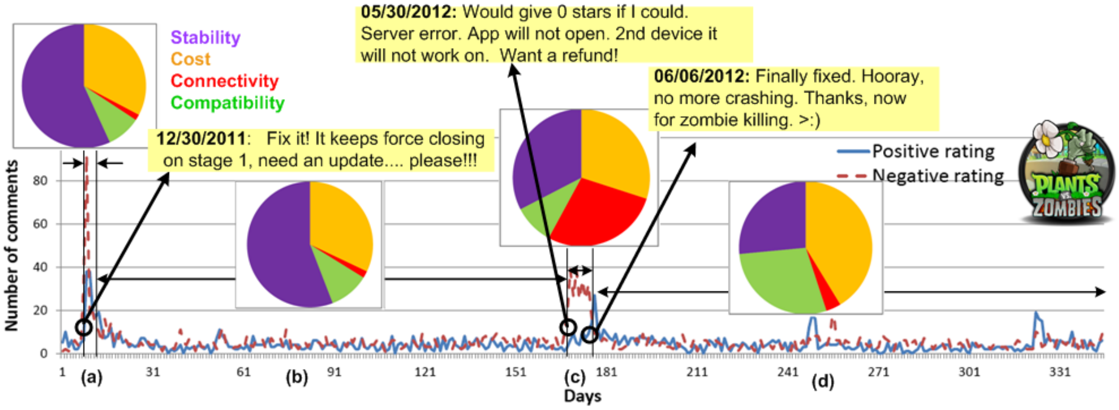
Other than static / dynamic analysis that extracts apps' behaviors from the source code or in runtime, there are other approaches from which apps' behavior can be determined. In a side project collaborated with Fu et al. [59], Natural Language Processing (NLP) technologies were applied to user reviews to diagnose problems associated with different apps. This was done in part using the Stanford Topic Modeling Toolbox [14] to train Latent Dirichlet Allocation models, resulting in a 10-topic model, as summarized in Table 4.

The topics are sorted by their average proportions across the distribution of all documents. We added a descriptive word to each topic at the top of Table 4 to represent the major concept each topic is talking about. Most topics exhibit clear reasons why users dislike an app. These reasons related to functional features such as picture and telephony, performance issues such as stability and accuracy, and other important factors such as cost and compatibility.

Looking at the 10 topics that had registered the most complaints, we can see that privacy is not included as such. It can be seen however that users complain about excessive behaviors such as spamming ads. This type of spamming very often comes with behavioral ad services collecting users' private information such as location and phone ID to deliver context-based advertisements. While this may not be apparent to many end-users, this is clearly an example of a privacy-invasive behavior directly related to the practices studied in this dissertation.

By performing topic modeling in different time windows, a dynamic historical view can be created to illustrate the time span of an app, highlighting the different distribution of user complaints about different problems. For example, two new version releases

[11] The content of this sub-chapter is published in KDD'13 [120].


05/30/2012: Would give 0 stars if I could. Server error. App will not open. 2nd device it will not work on. Want a refund!

06/06/2012: Finally fixed. Hooray, no more crashing. Thanks, now for zombie killing. >:)

12/30/2011: Fix it! It keeps force closing on stage 1, need an update.... please!!!

Figure 7 . We use time series to visualize the life story of Plants vs. Zombies, and topic analysis is performed for different segments of the time series.

naturally separated the user review time series of Plants vs. Zombies into four segments (See Figure 7). This game was first introduced to Google Play on December 21, 2011 (Day 1). There was a significant burst of negative reviews due to the instability of the initial release. Following this spike, stability remained the main source of complaints until a follow-on release in May 2012, which fixed the stability problem but resulted in connectivity issues. Approximately a week after this incident there was a spike of positive review on the time series plot, containing reviews such as *"Finally fixed. Horray, no more crashing… "*, indicating that the connectivity problem had been solved. This dynamic view provides a historical view of apps, which is extremely useful for users, developers and analyst to gain a deeper understanding of apps. By combining all reviews across the market, high-level market trends can also be identified to improve the market efficiency.

Similar NLP technologies can also help improve app analysis in terms of providing more context information. For example, Chen et al. studied the maturity rating [34] by performing text-mining on app descriptions and users' reviews. They developed mechanisms to verify the self-reported maturity rating of mobile apps and investigated possible reasons behind the incorrect ratings. Although slightly outside the scope of mobile privacy, their work demonstrated the ability and flexibility of NLP techniques to study the content of an app.

Along the same direction, through mining app description and user reviews, we can identify the services and functionality this application provides. Currently Google classified all the apps into 30 categories based on their functionalities; however, this taxonomy is still too coarse to infer whether certain private resources are necessary for certain apps. Text mining techniques can used to generate more comprehensive attributes to describe apps' functionalities.

3.5 Summary

In this chapter, we described the detailed procedures involved in downloading and analysis over 100K mobile apps. Specifically, we discussed how we use Androguard, an Android reverse engineering tool to perform static code analysis on apps, focusing on identifying what sensitive user data/resources are used and why. We leveraged the Amazon EC2 cloud to enable the batch processing to speed up the analysis of this large quantity of apps. To identify the purpose why sensitive user data are used, we looked up the top 400 3rd-party libraries that are most frequently used in all these apps and categorized them into 9 categories based on what type of services they provide. We also analyze how different types of resources (permissions) are used for various purposes. We further pointed out the potential of leveraging NLP techniques in app analysis.

4 IDENTIFYING PATTERNS IN APPS' PRIVACY BEHAVIORS

In the previous chapter, we discussed the techniques that we used to crawl Google Play and to perform code level static analysis of individual apps. To gain a deeper understanding of the Android mobile app market, and to identify typical patterns that apps consume users' private information, we applied machine learning techniques to identify common patterns in apps' privacy-related behaviors. More specifically, we want to see if there exist several groups of apps that exhibit particular characteristics in collecting and consuming users' sensitive data; thus we can discuss their privacy risks and coping mechanisms separately.

4.1 Preprocessing

We performed several preprocessing steps to code our dataset properly for the clustering task.

First, we organized the raw static analysis results by aggregating the 3rd-party library usage based on their categories (as mentioned in Table 2) and what Android permissions they use. For each permission *p* requested by an app *a*, we use one attribute to encode if *p* is triggered by internal use, and count how many 3rd-party libraries *l* in the category *c* is bundled in *a*. We only focused our analysis on the top 11 most sensitive and frequently used permissions, as identified earlier [49]. They are: INTERNET, READ_PHONE_STATUS, READ_CONTACT, GET_ACCOUNTS, BLUE_TOOTH, ACCESS_FINE/COARSE_LOCATION, SEND_SMS, READ_SMS, CAMERA, and RECORD_AUDIO (see Appendix A for the description of these permissions). By doing this, we can greatly reduce the number of features used to describe each app, and hence reduce the sparsity in our data. We also counted the number of URLs that app *a* is connecting to. In total, we used 131 attributes to represent the static analysis results for each of the 89,903 apps.

Secondly, we append the apps' meta-data to our dataset. These meta-data include the name of the app, developer, the range of the download number, the average rating of the app, star rating distribution, the number of user reviews, etc. Together with the app behavioral attributes, each app has a feature set of 144 features.

Thirdly, we perform a simple dimension reduction by eliminating the features that are constant or nearly constant. This dimension reduction results in a remaining matrix of 120 features.

Finally, we normalize the dataset such that all the features except the text fields have the same value range of [0.0, 1.0].

4.2 Clustering Algorithms and Distance Functions

We used hierarchical clustering with an agglomerative approach to cluster apps' privacy related behaviors, where each observation starts in its own cluster, and pairs of clusters are merged as one step moves up the hierarchy according to the distance measures and agglomerative algorithms. In the general case, the complexity of agglomerative clustering is $O(n^3)$ [89]. Though its time complexity is not as fast as k-means or other flat clustering algorithms, we chose hierarchical clustering mainly due to three reasons: (1) it is flexible in its selection of distance functions, which gives us ample room to try out different distance functions since we did not know which one would work best; (2) the hierarchical structure produced by hierarchical clustering is much more informative than the unstructured clusters, hence the clustering results are more likely to be interpretable and less likely to result in artificial boundaries (such as those sometimes produced by centroid-based clustering techniques like k-means); (3) it does not require us to pre-specify the number of clusters (in contrast to k-means) and the results are deterministic (stable). In short, we intentionally sacrifice efficiency for the sake of obtaining clusters that are more likely to capture genuine differences between apps and more likely to be interpretable.

In our work, given the new dataset and new problem, we first explored possible distance measures and agglomerative methods. More specially we ran our hierarchical clustering algorithms with the following distance measures [89]:

A. **Euclidean distance (ECL)**:
Euclidean distance between two points in a 2D space is given by the Pythagorean formula. When extended to n-dimensional space, inner products are used. In Cartesian coordinates, if $p=(p_1,p_2,...p_n)$ and $q=(q_1,q_2,...q_n)$ are two points in Euclidean n-space, then the distance from p to q, or from q to p is given by:

$$d(p,q) = d(q,p) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + ... + (q_n - p_n)^2} = \sqrt{\sum_{i=1}^{n} (q_n - p_n)^2}$$

B. **Manhattan distance (MHT)**:
Manhattan distance measures the distance between two points as the sum of the absolute differences of their Cartesian coordinates, also known as rectilinear distance, *L1* distance or taxicab metric. More formally,

$$d(p,q) = d(q,p) = \| p - q \| = \sum_{i=1}^{n} | p_i - q_i |$$

C. **Canberra distance (CBR)**:
Canberra distance is a weighted version of Manhattan distance, which has been used as a metric for comparing ranked lists and for intrusion detection in computer security. The Canberra distance between vector p and q in an n-dimensional real vector space is given as follows:

$$d(p,q) = d(q,p) = \sum_{i=1}^{n} \frac{| p_i - q_i |}{| p_i | + | q_i |}$$

D. **Binary distance** (**BNR**):
Binary distance (so called Hamming distance) measures the minimum number of substitutions required to change one binary vector to another. In our case, we coded non-zero entries as '1's to form a binary matrix. So for binary vectors p and q the hamming instance is equal to the number of ones (population count) in *p XOR q*.

In addition, we explore the following agglomerative methods in our experiments:

A. **Ward’s Method (WAR):**
Ward's method offers a general agglomerative hierarchical clustering procedure, where the criterion for choosing the pair of clusters to merge at each step is based on the objective function that finds the *minimum between-cluster distance*. In other words, the pair of clusters with the closest boundaries are merged.

B. **Centroid Method (CTD)**:
Centroid method suggests merging the two clusters with *minimum distance between their centroids*. The centroid of each cluster is usually defined by averaging all the points within the cluster.

C. **Average Linkage (AVG)**:
The average linkage method merges clusters based on their average distances. It computes the distances between the pairs of points in two clusters and takes the mean of all these pairs as the average distance between two clusters. The two clusters with the minimum average distance are merged.

D. **McQuitty’s Similarity (MCQ)**:
McQuitty’s method merges together the pair of clusters that have the highest average McQuitty’s similarity value, as defined in [100]. This agglomerative method has been proven to be effective in text clustering.

We limited our exploration to the above-mentioned distance functions and agglomerative methods, since other distance functions or agglomerative methods either produce similar results as the above-mentioned ones or are not appropriate for our tasks based on the characteristics of our data. As research on clustering techniques continues, it is possible that new techniques could provide even better results than the ones we present. We found however that by themselves these techniques were already sufficient to isolate very different categories of mobile apps, when it comes to their permissions and the purposes associated with these permissions.

4.3 Evaluating Clustering Algorithms

To select the best agglomerative method and the best distance function for our problem, we experimented with various ways of combining the four agglomerative methods and four distance measures by using the R package “hclust” [102]. We conducted all the experiments on a Linux machine which has *XeonE5-2643 3.3GHz CPU* (16 cores) and 32G memory. We selected the most popular 20,000 apps with all encoded features as the data input to perform the clustering analysis.

We have two selection criteria in determining which combination of distance function and agglomerative method to use. First, the combination should not produce clusters with extremely skewed structures in dendrograms. A dendrogram is a tree diagram frequently used to illustrate the arrangement of the clusters produced by hierarchical clustering, where x axis represents all the instances in the vector space, and the y axis represents the range of distances in this vector space. The tree structure in the dendrogram illustrate how clusters merged at each iteration. We check this by manually inspecting the dendrograms produced by the clustering. The other criteria are three internal measures, namely connectivity, Silhouette Width and Dunn Index, which validate the clustering results based on their connectivity, compactness and degree of separation.

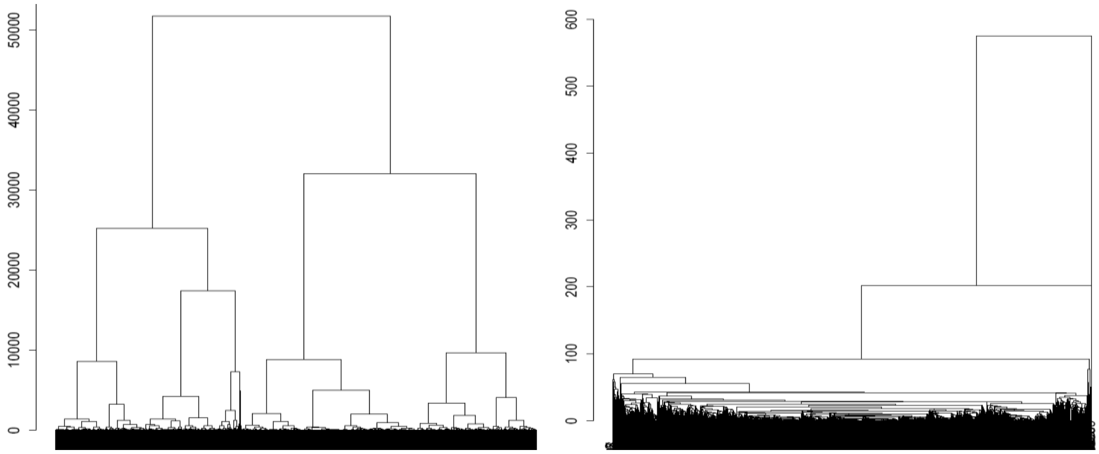
Figure 8 shows two examples of dendrograms produced by different agglomerative methods and distance functions. The left one presents a fair structure in the sense that the clusters in the higher hierarchy include a good number of instances, whereas the right dendrogram presents a skewed structure --- even in the top several levels of the hierarchical structure, the clusters only contain a couple of instances. After inspecting all the resulting dendrograms, we eliminated the following combinations: Average Linkage methods with all distance measures, binary distance with Ward's method and centroid method, Manhattan distance with McQuitty's method, and Euclidean distance with McQuitty's method, resulting in 8 remaining potential distance and agglomerative method combinations.

In the second step, we use three internal measures (provided by R package "clValid"[101]) to quantitatively evaluate the remaining combinations. These measures reflect the *compactness, connectivity* and the *separation* of the cluster partitions.

Connectivity measures to what extent observations are placed in the same cluster as their nearest neighbors in the data space, and is measured as [65]:

$$Conn(C) = \sum_{i=1}^{N} \sum_{j=1}^{L} x_{i,nn_{i(j)}}$$

Where *nn* $_{i(j)}$ denotes the *j* th nearest neighbor of observation *i*, and let $x_{i,nn_{i(j)}}$ be zero if *i* and *j* are in the same cluster and *1/j* otherwise, and *L* is a parameter giving the number of nearest neighbors to use (*L* is set to 10 in our case). The connectivity has a value between zero and ∞ and should be ***minimized***.



(a) Euclidean distance with Ward method

(b) Manhattan distance with Average Linkage method

Figure 8: Two examples of dendrograms produced by different agglomerative methods and distance functions. The left one is produced by applying Ward's method with Euclidean distance. The right one is produced by applying Average Linkage method with Manhattan distance. The hierarchical structure is very skewed even at the top level.

Compactness assesses cluster homogeneity, usually by looking at the intra-cluster variances, while separation quantifies the degree of separation between clusters (usually by measuring the distance between cluster centroids). Popular methods combine the two measures into a single score, such as the ***Dunn Index*** [42] and ***Silhouette Width*** [105].

The Dunn Index is the ratio of the smallest distance between observations not in the same cluster to the largest intracluster distance, computed as

$$D(C) = \frac{\min_{C_k, C_l \in C, C_k \neq C_l} (\min_{i \in C_k, j \in C_l} dist(i, j))}{\max_{C_m \in C} diam(C_m)}$$

where $diam(C_m)$ is the maximum distance between observations in cluster C_m. The Dunn Index has a value between zero and ∞ and should be ***maximized***.

The Silhouette Width is the average of each observation's Silhouette value which measures the degree of confidence in the clustering assignment of a particular observation. For observation *i*, it is defined as

$$S(i) = \frac{b_i - a_i}{\max(b_i, a_i)}$$

where a_i is the average distance between *i* and all other observations in the same cluster, and b_i is the average distance between *i* and the observations in the "nearest neighboring cluster". The Silhouette width thus lies in the interval [-1,1] and should be ***maximized***.

We varied the number of clusters *k* from 2 to 20 to create the cluster labels for each of the 8 remaining distance and agglomerative method combinations, and then rank them based on the three internal measures respectively. Table 5 summarizes the rankings based on internal measures.

It shows that Canberra distance with Ward's method when *k=5* has the highest Silhouette width (should be maximized) and Dunn Index (should be maximized), and it ranks the second for the connectivity (should be minimized). Collectively, we choose the clusters produced by this setting and present the visualization and interpretation in the following sections.

	Connectivity		**Dunn Index**		**Silhouette Width**	
Rank	Dist- aggl-k	Value	Dist- aggl-k	Value	Dist- aggl-k	Value
Top 1	ECL-WAR-5	4.38	CBR-WAR-5	0.40	CBR-WAR-5	0.98
Top 2	CBR-WAR-5	4.96	ECL-WAR-9	0.26	BNR-MCQ-4	0.86
Top 3	MHT-WAR-6	5.34	CBR-WAR-6	0.23	MHT-WAR-5	0.81

Table 5: Top 3 clustering configurations for each internal measure. Clusters obtained by using Canberra distance and Ward's method with *k*=5 (CBR-WAR-5) ranks first in Dunn Index (should be maximized) and Silhouette Width (should be maximized) and ranks second in the connectivity (should be minimized). We select this configuration as its best performance overall.

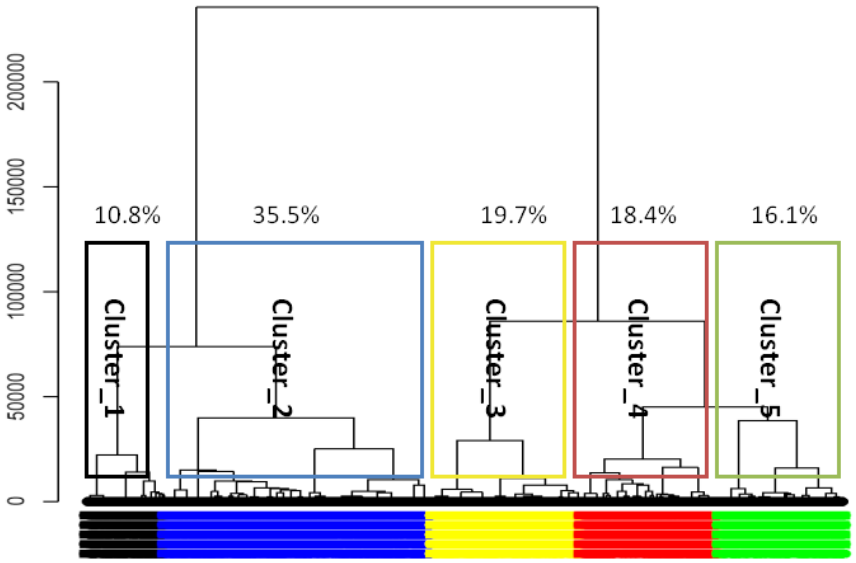


Figure 9: The dendrogram of hierarchical clustering with Canberra distance and Ward's method. It visualizes how clusters merge in each iteration. The five different colors at the bottom represents five different cluster labels assigned to all the instances when *k=5*.

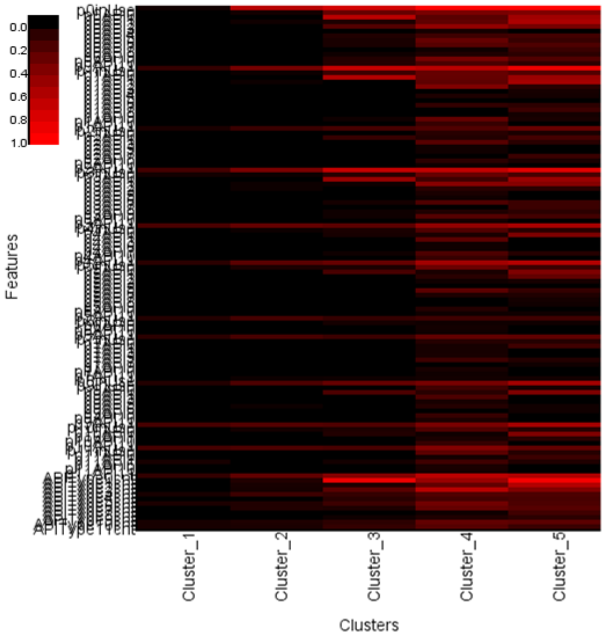


Figure 10: A heat map plots the centroid of each cluster. The brighter the color represents the higher values in corresponding attributes. We can see distinguishing patterns in all the five clusters.

4.4 Resulting Clusters

We plot the dendrogram of the iterations of hierarchical clustering with Canberra distance and Ward's method in Figure 9. At each iteration, cluster merging is represented

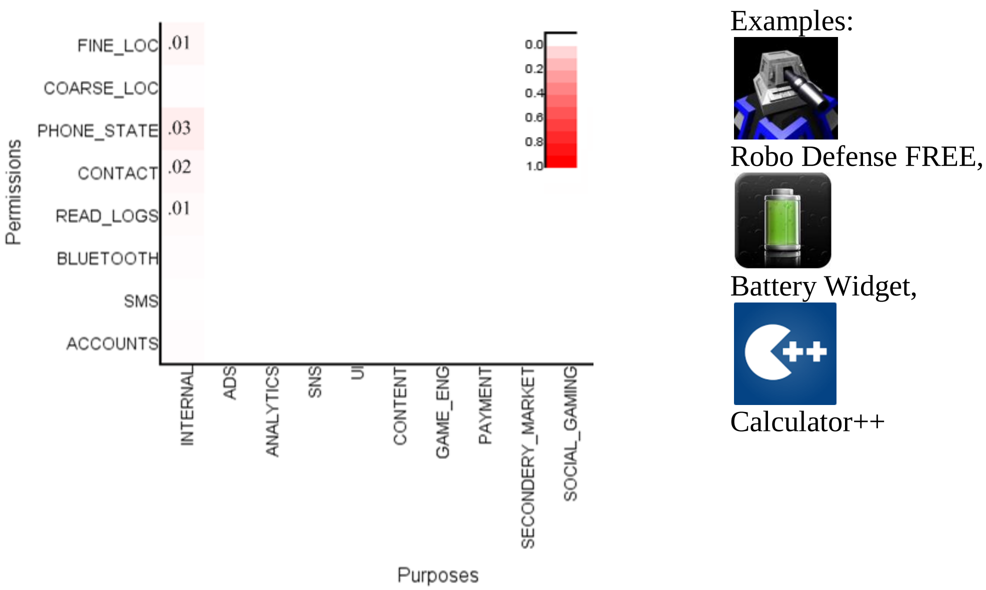


Figure 11: Heap map visualization of the centroid of cluster_1. Permissions are displayed along the vertical axis, while the possible purposes associated with these permissions are displayed along the horizontal axis Apps in this cluster seldom use any sensitive permissions. More red color indicates a higher proportion of apps requesting a given permission for a particular purpose.

by a merge in the hierarchical structure. Five different colors at the bottom represent five different clusters resulted in this process. The smallest cluster (in black) contains 10.8% of apps and the largest cluster (in blue) contains 35.5% of apps.

To get a first impression of the resulting clusters, we compute the centroid of each cluster by averaging the attributes of all the instances in the same cluster. We use a heat map to visualize the centroids of all the five clusters in Figure 10, where the brighter the color is the larger the value in the corresponding attributes[12]. Although human perception can easily tell there are significant differences among the five clusters in the visualization, it is not straightforward to spot the distinct characteristics of each cluster.

To better understand each cluster and its privacy implications, in the following sub-sections, we separately plot the five clusters in 2 dimensional grid representations, where the vertical dimensions represent the different usages of permissions and the horizontal dimensions represent why (purpose) of using certain permission. The number in each grid roughly translates to the portion of apps used specific permission for a specific purpose. We will also discuss the potential privacy risks of each cluster of apps.

4.4.1 Cluster_1: Few Requested Permissions

Cluster 1 is the smallest cluster among the five. It contains just 10.8% of apps. Figure 11 depicts the centroid of cluster 1. We can see that only a few entries are filled with very

[12] Since the dataset is normalized before clustering, all the entries have values within the range of [0.0, 1.0]. It is hard to assign a physical meaning to the normalized value, though roughly the larger the value means more frequent usage of certain sensitive resources for certain purposes.

light red. This suggests that this cluster of apps seldom use permissions that involving sensitive user data. A lot of them are utility apps such as calculator, battery widget, or simple games such as Robo Defense FREE. Because of the absence of permission usage, this type of cluster poses almost no privacy risks to users.

4.4.2 Cluster_2: Permissions Primarily Requested for Internal Use

The second cluster of apps is the largest (35.5% of all apps we crawled). The apps in this cluster consume permissions for their functionality (internal use) most of the time, as shown in Figure 12. Note that, the "internal use" here does not necessarily imply that user data never leaves the mobile devices, but rather it refers to the situation where the sensitive user data are not requested by any 3rd-party libraries bundled with the application.

An example of an app in this cluster is Google Maps, which uses the user's location information primarily to support location-based search and navigation functionality. Another example is WhatsApp Messenger, which accesses the phone book (READ_CONTACT) to facilitate messenger service. Yet another app in this category is YouTube, which accesses a user's account information but solely as part of its authentication process. This category of apps seldom uses 3rd-party libraries and uses users' sensitive data primarily within the app's native code. The privacy risk associated with these apps appears to be low, which is similar to the first cluster. While developers or companies who produce apps in this cluster access and transmit sensitive user data to their servers, they seem to do so primarily to support their app's core functionality. Obviously, once the data leave the mobile device, we have no way of knowing exactly how it will be used. In light of the many third party libraries found in the code of other apps, it is reasonable to assume that most of these apps probably do not share this data: if they were, they would likely do so using 3rd party APIs embedded in their app's code, but there are probably some exceptions. A privacy policy could possibly help further clarify whether any such sharing takes place at the server level. In short, while this category of apps seems benign, they might still pose privacy risks, especially if the data being collected does not seem to be required by the app's core functionality.

4.4.3 Cluster_3: Ad-powered apps

In the mobile ecosystem, a large portion of app developers monetize their products by bundling behavioral ad libraries [31]. This cluster of apps (Figure 13) demonstrates this intention clearly, since in most cases permissions are used for delivering targeted ads. Multiple types of personal information, including users' location, phone number, and contact list could be used to profile users' life style, habits and interests. Since an ad library might be bundled in multiple apps, the ad agency could potentially aggregate the data collected through multiple apps to build a more comprehensive profile of users. In short, the privacy risks of this type of apps is not from developers, since they do not receive sensitive information, but rather from the companies and agencies involved in collecting this data, building profiles and delivering targeted ads.

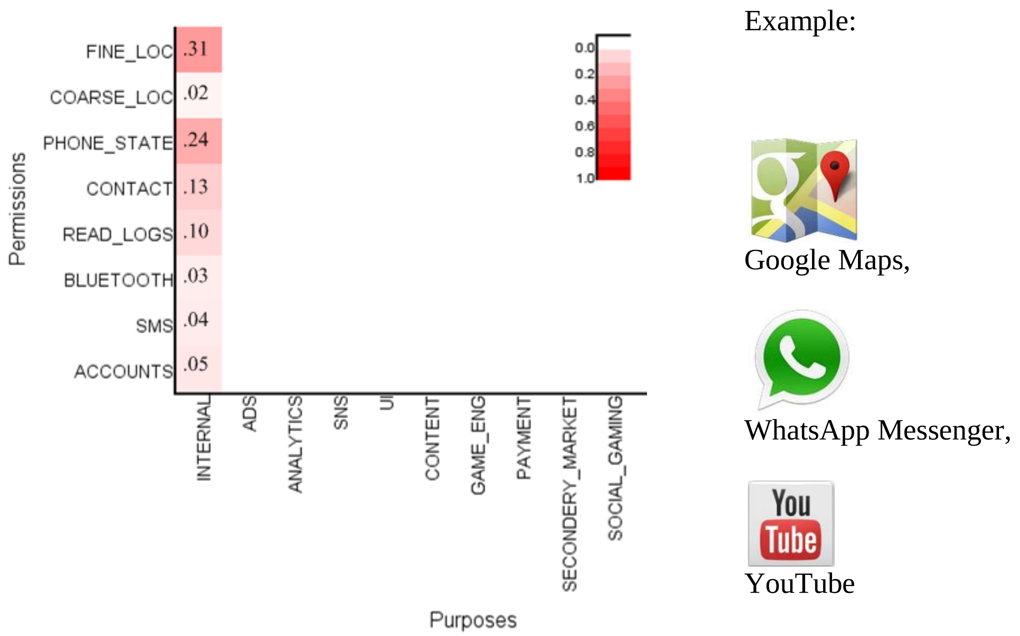


Figure 12: Heap map visualization of the centroid of cluster 2. This cluster of apps uses sensitive permissions mostly for functionality purposes.

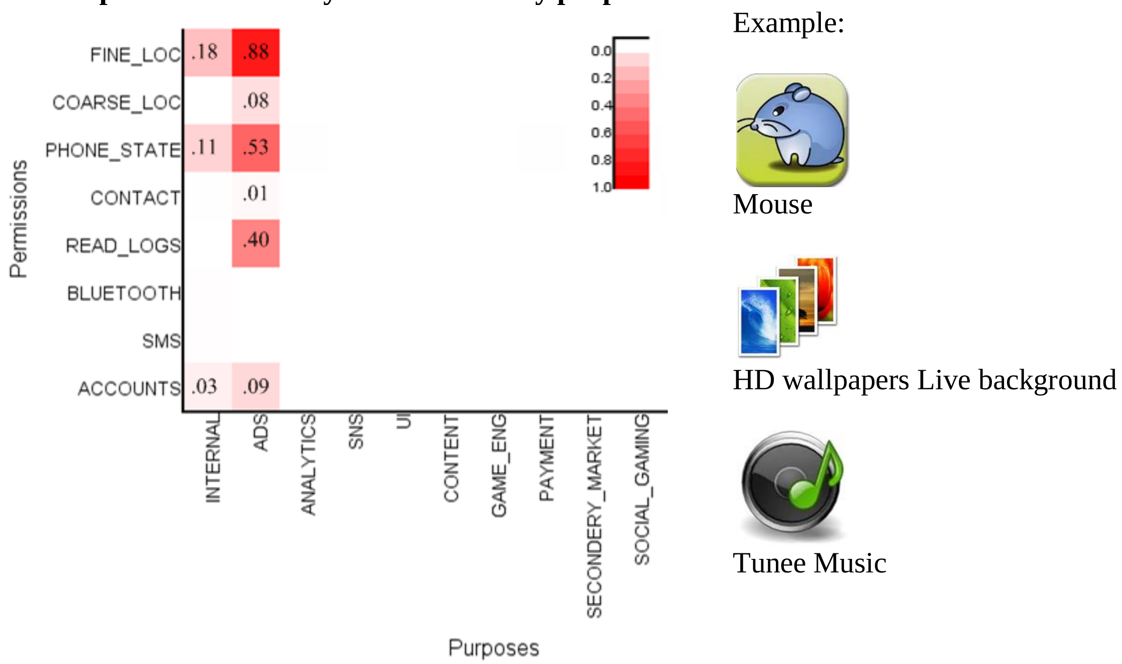


Figure 13: Heap map visualization of the centroid of cluster 3. This type of app uses sensitive permissions because of the targeted ads libraries that bundled inside them.

Some work has sought to block behavioral ad libraries to prevent excessive data collection [29, 68]. It is unclear to what extent such an approach will remain viable in the long run. This is because a large portion of app developers rely on behavioral advertising to monetize their products, especially those who develop free apps. Blocking ad libraries would greatly discourage these developers and blunt the innovation we have seen in smartphone apps. A more practical approach would involve the development of an infrastructure or functionality that exposes these practices to users and enable them to make better informed decisions. Over the past year, we have seen many big players in mobile advertising (such as Admob, Tapjoy, InMobi, etc.) refining their privacy policies. Some initiatives have also been launched that aim to better control the collection and use of user data. These efforts include an agreement between the California Attorney General and six major mobile app marketplaces (including GooglePlay, iTunes and the Amazon App store) to require app developers to include privacy policies. They also include initial self-regulatory efforts by industry. Whether these efforts will be sufficient to empower end-users to make informed decisions remains to be seen.

4.4.4 Cluster_4: Apps that Promote Sharing

Compared to other clusters, this group of apps (see Figure 14) is often bundled with SNS libraries. These libraries are usually used to let users share app-related content on social network sites, like Twitter and Facebook, together with other contextual information such as location. By promoting the interaction with SNSs, these apps can achieve two goals. One is that the app itself can act as a portal for users to manage their social networks. The other is to leverage the SNSs as dissemination channels to propagate and advertise this app. Together with other facets like targeted advertising as well as in-app purchasing, either way could make the app more popular.

The privacy risks associated with this type of app are similar to those posed by social network sites. Users are responsible for the consequences of sharing app-related information and potentially their own information through SNS libraries. This type of sharing might require users to balance factors like social capital, maintaining social images of themselves, and protecting their privacy. At the same time, when combined

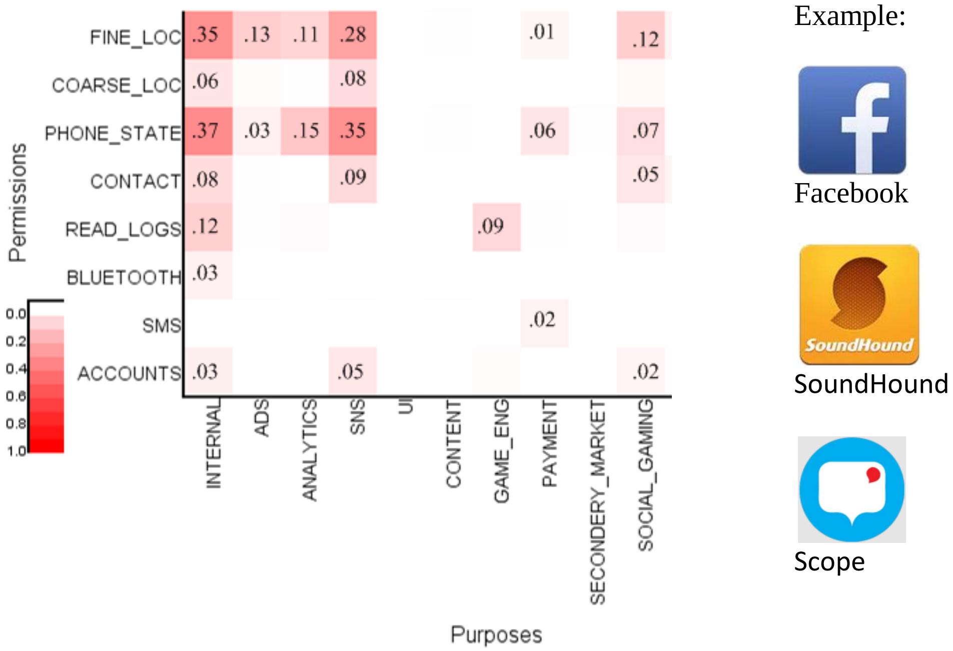


Figure 14: Heat map visualization of the centroid of cluster 4. This cluster of apps consume permissions most for internal functionality and for promoting sharing through bundled social network libraries.

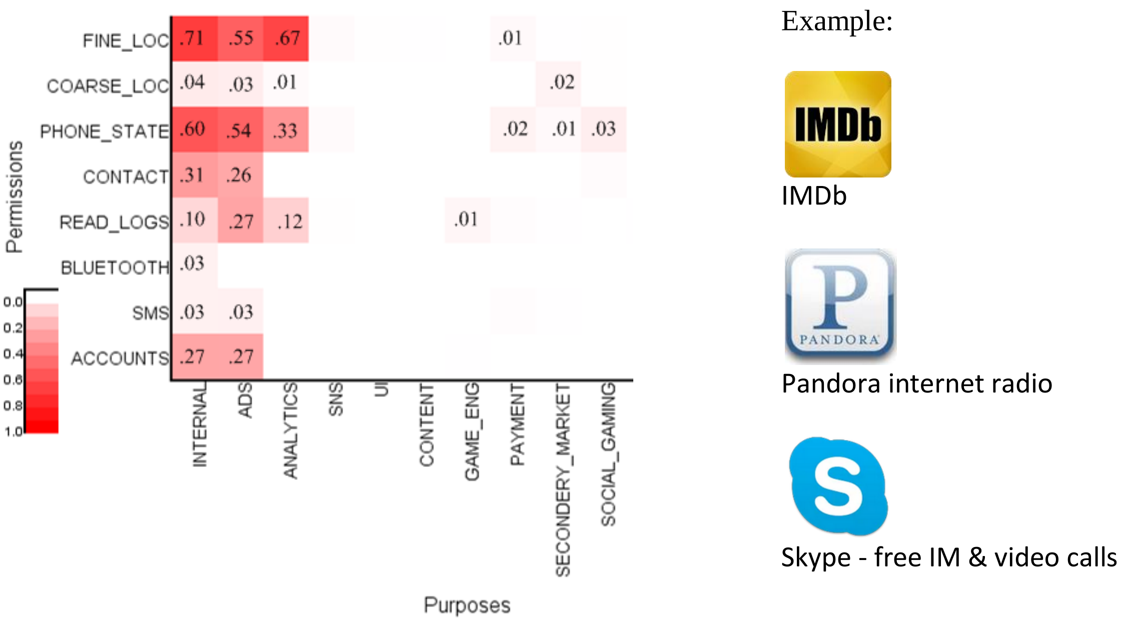


Figure 15: Heat map of cluster 5. This cluster of apps consumes permissions both internally and for delivering targeted ads and facilitating mobile analytics.

with internal functionalities or for targeted advertising, this type of app also suffers the same privacy risks as the previous two clusters.

4.4.5 Cluster_5: Multi-purpose Apps

The last cluster of apps (Figure 15) is the most sophisticated cluster, which contains approximately 16.13% of the free apps we crawled from Google Play. This type of apps uses permissions for multiple purposes and is more likely to be produced by developers who are interested to know how users use their apps, since we also observe a strong penetration of mobile analytics libraries such as Flurry and Localytics. Mobile analytics libraries usually provide paid services to app developers to help them understand how consumers interact with their apps. The information they collect might also include personal information such as users' location, phone number, etc.

Similar to targeted advertising libraries, mobile analytics companies can harvest user data from multiple apps, and are hence able to build more comprehensive user profiles. We also see the trend that some mobile analytic companies have started to offer targeted advertising in the last two years. It is a reasonable assumption that the user profiles built from mobile analytics are also being used to assist their ad services. Combined with other sensitive data usage by apps themselves and targeted advertising, this cluster of apps poses more convoluted privacy risks to users.

4.5 Discussion

We identified patterns in privacy-related behaviors of apps by clustering mobile apps in terms of how and why they use different private data. In this sub-section, we discuss how these findings help us move forward in our mobile app privacy research.

First of all, the five clusters of apps we identified provide a new way to organize mobile apps, and help us understand how these typical patterns are distributed in the app market. In contrast to previous work by Frank et al. [56] which focused on identifying app-permission patterns, namely groups of apps with similar permission patterns. Our analysis provided a more in-depth look at these patterns by attempting to also capture the purpose associated with different app-permissions. By distinguishing between ten different purposes associated with different app permissions, we are able to develop a deeper understanding of the information flows associated with different app-permissions. As we show in the next chapter, the purpose associated with an app-permission pair has a significant impact on people's privacy preferences when it comes to deciding whether they feel comfortable granting a given permission to an app. Specifically, in the next chapter, we describe a crowdsourcing study where we asked participants why they believe an app is requesting a given permission and to what extent they feel comfortable granting that permission.

As shown in the next chapter, different clusters of apps give rise to different privacy concerns. Given the different ways in which they use sensitive data, these clusters of apps induce different types of privacy risks. In other words, permissions by themselves are not sufficient for users to make decisions. Instead, additional details about the purpose associated with the collection of sensitive data are critical. Such information should minimally be provided in the form of privacy policies, though, as has been shown by others [91], users are unlikely to read these policies. In the following chapter, we show how models of people's privacy preferences can also be used to inform the design of privacy displays that highlight those issues that are most likely to impact people's privacy decisions.

Finally, though this clustering information might not be intuitive for end users, it can be treated as another attribute to describe apps that could potentially provide extra value to other stakeholders. For example, these attributes could be used for services such as mobile app recommender systems, or can be used as a clue to assign privacy scores to individual apps.

This chapter demonstrates the exploration and the knowledge we discovered with regard to apps' privacy-related behaviors. We also produce a valuable dataset that describes apps in terms of what sensitive user data they consume and why. We believe that by applying more advanced machine learning techniques or mining this dataset from other angles, we can uncover more facts, patterns, and knowledge about either individual mobile apps or the mobile market as a whole.

5 NOTICE & AWARENESS: HOW TO INFORM USERS?[13]

The FTC has identified five core principles (Fair Information Practice Principles) to protect consumers' privacy, among which "Notice/Awareness" is the most fundamental one [58]. This principle states *"Consumers should be given notice of an entity's information practices before any personal information is collected from them."* However, multiple user studies have found that mobile app users seldom pay attention to permission screens and have a hard time understanding their privacy implications [54, 77]. One major contribution of this thesis is to develop better ways informing users of pertinent information in a more effective and understandable way.

5.1 What to Show?

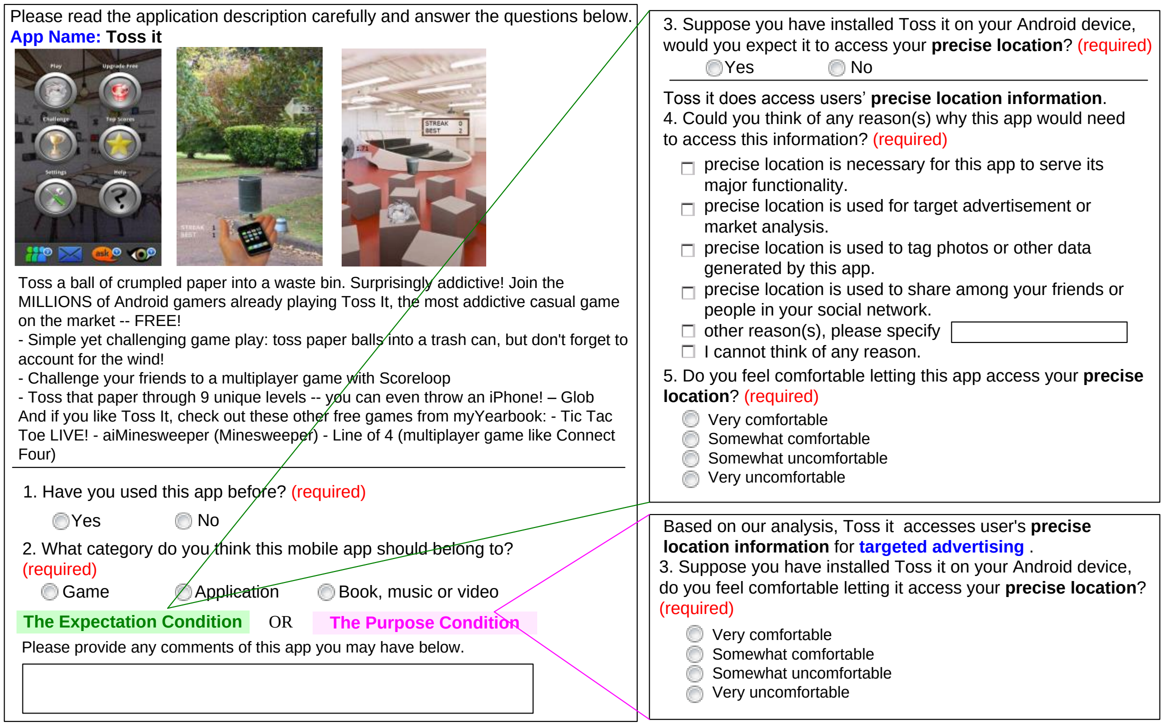
The existing permission screens of Android generally lack adequate explanation and definitions, which motivated us to explore what information should be conveyed to users that can help them better understand the privacy implications of apps' sensitive data usages.

One thing we learned from previous location sharing privacy studies is that users have distinct privacy preferences for different kinds of sharing [110]. This makes sense in the context of mobile app privacy as well. People's perceptions of whether an action is reasonable, or how that action makes users feel with respect to their privacy are greatly influenced by why an action is taken. For example, is a given app's use of one's location appropriate or not? It all depends on the ***purpose***: for a blackjack game with no clear reason for collecting location information, probably not, but for a map application to provide point-to-point navigation, very likely so. Therefore, a clear explanation as to why the sensitive data is required is necessary to properly inform users. Thanks to a number of research projects, there are many existing application analysis tools [36, 45-47, 55] that we can leverage to identify the purposes of the data disclosure.

In addition, we frame mobile privacy in the form of people's ***expectations*** about what an app does and does not do, focusing on where an app breaks people's expectations. There has been a lot of discussions about expectations being an important aspect of privacy [109]. We framed our inquiry on Norman's notion of *mental models* [94]. All people have a simplified model that describes what they think an object does and how it works (in our case, the object is an app). Ideally, if a person's mental model aligns with what the app actually does, then there would be fewer privacy problems since that person is fully informed as to the app's behavior. However, in practice, a person's mental model is never perfect. We argue that by allowing people to see the most common misconceptions about an app, we can rectify people's mental models and help them make better trust decisions regarding that app.

The notion of expectations is fairly common in discussions of privacy [109]. For example, the famous 1967 US Supreme Court case Katz v United States ruled that people

[13] Part of this chapter has been previously published in Ubicomp'12 [82], and other parts included in a paper submitted to CHI'14 [120].



Please read the application description carefully and answer the questions below.
App Name: **Toss it**

Toss a ball of crumpled paper into a waste bin. Surprisingly addictive! Join the MILLIONS of Android gamers already playing Toss It, the most addictive casual game on the market -- FREE!
- Simple yet challenging game play: toss paper balls into a trash can, but don't forget to account for the wind!
- Challenge your friends to a multiplayer game with Scoreloop
- Toss that paper through 9 unique levels -- you can even throw an iPhone! – Glob
And if you like Toss It, check out these other free games from myYearbook: - Tic Tac Toe LIVE! - aiMinesweeper (Minesweeper) - Line of 4 (multiplayer game like Connect Four)

1. Have you used this app before? (required)

○ Yes ○ No

2. What category do you think this mobile app should belong to? (required)

○ Game ○ Application ○ Book, music or video

The Expectation Condition OR The Purpose Condition

Please provide any comments of this app you may have below.

3. Suppose you have installed Toss it on your Android device, would you expect it to access your **precise location**? (required)

○ Yes ○ No

Toss it does access users' **precise location information**.
4. Could you think of any reason(s) why this app would need to access this information? (required)

□ precise location is necessary for this app to serve its major functionality.
□ precise location is used for target advertisement or market analysis.
□ precise location is used to tag photos or other data generated by this app.
□ precise location is used to share among your friends or people in your social network.
□ other reason(s), please specify
□ I cannot think of any reason.

5. Do you feel comfortable letting this app access your **precise location**? (required)

○ Very comfortable
○ Somewhat comfortable
○ Somewhat uncomfortable
○ Very uncomfortable

Based on our analysis, Toss it accesses user's **precise location information** for **targeted advertising** .
3. Suppose you have installed Toss it on your Android device, do you feel comfortable letting it access your **precise location**? (required)

○ Very comfortable
○ Somewhat comfortable
○ Somewhat uncomfortable
○ Very uncomfortable

Figure 16: Sample questions to capture users' mental models. Participants were randomly assigned to one of the conditions. In the *expectation condition*, participants' were asked to specify their expectations and speculate about the purpose for this resource access. In *the purpose condition*, the purpose of resource access was given. In both conditions, participants were asked to rate how comfortable they felt having the targeted app access their resources.

could not have their telephone calls monitored without a warrant because there was a "reasonable expectation of privacy" [15], with this famous phrase being the basis of a test used by US courts to evaluate the reasonableness of legal privacy protections. Our notion of *privacy as expectations* is a different construct, focusing primarily on people's mental models of what they think an app does and does not do. Our core contribution is in operationalizing privacy in this manner of capturing people's expectations as well as reflecting other people's expectations directly in a privacy summary to emphasize places where an app's behavior did not match people's expectations.

5.2 How to Gather Data?

As mentioned in the previous section, the purpose of disclosing private data can be identified by using various analysis tools [36, 45-47, 55]. By leveraging the power of cloud, app analysis can be easily scaled up to handle dissecting thousands or even tens of thousands of apps. On the other hand, traditional ways of collecting user feedback, such as interview or lab studies seems inadequate to catch up with the scale of data that we intend to collect. How to collect user feedback in an efficient and affordable way becomes a major challenge.

Inspired by work like [57, 90], we turn to crowdsourcing for help. There are four reasons why crowdsourcing is a compelling technique for examining privacy. Past work has shown that few people read End-User License Agreements (EULAs) [61] or web privacy policies [72], because (a) there is an overriding desire to install the app or use the web site, and reading these policies is not part of the user's main task (which is to use the app or web site), (b) the complexity of reading these policies, and (c) a clear cost (i.e. time) with unclear benefit. The current Android permission screen suffers from the same problems: (a) user's main task is to install the apps and this task is interrupted by going through permission screen. (b) the permission description text is lengthy and hard to understand. (c) Without fully understand its privacy implication, there is no clear benefit for users to go through these permission lists.

Crowdsourcing nicely addresses these problems. It dissociates the act of examining permissions from the act of installing apps. By paying participants, we make reading these permissions part of the main task and also offer clear monetary benefit. Lastly, we can reduce the complexity of reading Android permissions by having participants examine just one permission at a time rather than all of the permissions, and by offering clearer explanations of what the permission means.

5.3 Study Description

We recruited participants using Amazon's Mechanical Turk (AMT). We designed each Human Intelligence Task (HIT) as a short set of questions about a specific Android app and resource pair (see Figure 16). Participants were shown one of two sets of follow-up questions. One condition (referred to as *the expectation condition*) was designed to capture users' perceptions of whether they expected a given app to access a sensitive resource and why they thought the app used this resource. Participants were also asked to specify how comfortable they felt allowing this app to access the resource using a Likert scale that ranged from very comfortable (+2) to very uncomfortable (-2).

In the other condition (referred to as *the purpose condition*), we wanted to see how people felt when offered more fine-grained information. Participants were told that a certain resource would be accessed by this app and were given specific reasons for the access. We identified these reasons by app analysis and knowledge about ad networks. Participants were then asked to provide their comfort ratings as in the expectation condition. Finally, participants from both conditions were encouraged to provide optional comments on the apps in general. The separation of the two conditions allowed us to compare users' perceptions and subjective feelings when different information was provided.

We focused our data collection on four types of sensitive resources (as suggested by AppFence [68]): unique device ID (READ_PHONE_STATE), contact list (READ_CONTACT), network location (ACCESS_COARSE_LOCATION), and GPS location (ACCESS_FINE_LOCATION). We also restricted the pool of apps to the Top 100 most downloaded mobile apps on the Android market. The list of apps and their relevant permissions can be found in the Appendix B. Overall, 56 of these apps requested access to unique phone ID, 25 to the contact list, 24 to GPS location, and 29 to Network Location. This resulted in 134 app and resource pairs, i.e. 134 distinct HITs. For each HIT, we recruited 40 unique participants to answer our questions (20 per condition). We

MSE	Network Loc	GPS loc	Contact List	Unique ID
expectation [0,1]	0.0354	0.0303	0.0353	0.0363
comfort level [-2,+2]	0.7081	0.8136	0.6749	0.3067

Table 6: Crowd workers and experts have similar expectations toward targeted mobiles. In general, experts were slightly more skeptical about these privacy-related behaviors. Numbers in this table indicate the differences between the rating we obtained from the crowd workers and the experts, measured by the Mean Square Error.

limited our participants to Android users in U.S. and ensured a between-subjects design through a qualification test.

All the HITs of this study were completed over the course of six days. We collected a total of 5684 responses. 211 were discarded due to incomplete answers, and 113 were discarded due to failing the quality control question, yielding 5360 valid responses. There were 179 verified Android users in our study, with an average lifetime approval rate of 97% (SD=8.79%). On average, participants spent about one minute per HIT (M=61.27, SD=29.03), and were paid at the rate of $0.12 per HIT.

5.3.1 Feasibility of Using Crowdsourcing to Study Privacy

Though we already adopted quality control questions and qualification tests to ensure the validity of the data collected, we want to prove quantitatively that the crowdsourcing approach would not bias the results in gathering users' subjective feedback. To this end, we recruited five Android experts[14] to come to our lab; then we presented them with the same questions in the expectation condition and asked them to complete the questions for every resource and app pair (i.e., 134 sets of questions in total). We used the Mean Square Errors (MSE) to measure the differences between the subjective feedback

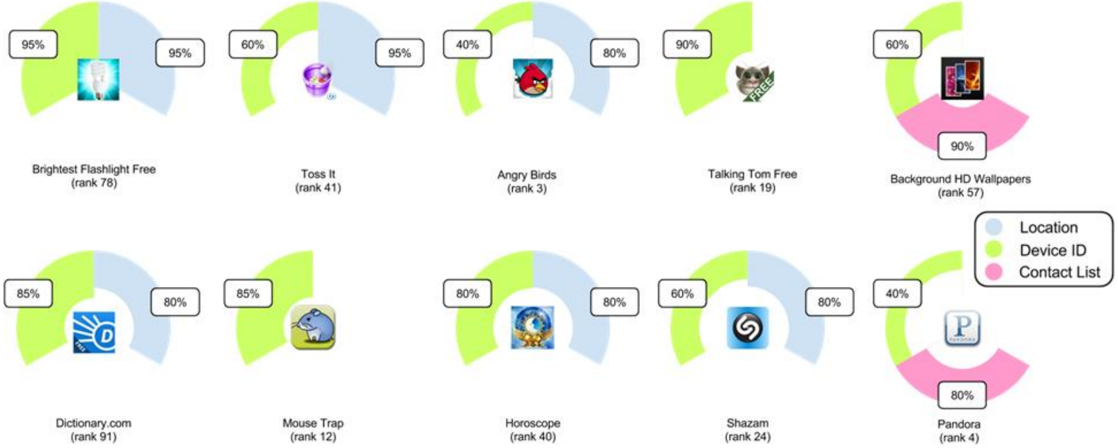


Figure 17. The percentage of users surprised about popular mobile apps using users' location, phone ID and contact list. This figure shows the top 10 apps with the least expected permission (among the top 100 most downloaded free Android Apps.)

[14] Someone with security background and has development experience in Android OS.

collected from crowd workers and experts (see Table 6). In general, crowd workers had a similar level of expectations as experts (i.e., MSE < 0.05). Experts on average appeared to be more skeptical about privacy-related behaviors of apps, which attributed to the slightly higher MSEs seen in the second row. Given the comfort level scaling from -2 to +2, these MSEs were still considered acceptable. In other words, these results demonstrate the validity and feasibility of crowdsourcing as a method to collect users' subjective feedback to study privacy.

We also wanted to see how previous experiences with an app impacted participants' expectations and level of comfort. To answer this question, we compared the responses between participants who had and hadn't used the app before. Our results show that the differences were not statistically significant with respect to their reported expectation and comfort rating of sensitive resource access. This finding suggests that people who use an app do not necessarily have a better understanding of what the app is actually doing, in terms of accessing their sensitive resources. It also suggests that, if we use crowdsourcing to capture users' mental models of certain apps, we do not have to restrict our participants to people who are already familiar with these apps, allowing us access to a potentially larger crowd.

5.3.2 How Users Feel about Popular Apps

To give a more intuitive impression of users' subjective feelings towards mobile apps, we first present the responses we collected with regard to the some popular apps that readers might familiar with. In Figure 17, we show the percentage of participants who were surprised by these popular mobile apps access users' location, unique phone ID and contact list. Figure 17 shows the data related to the top 10 apps with the least expected permission (among the top 100 most downloaded free Android Apps). From this figure we can see that even some very popular apps developed by well-known companies are harvesting more than necessary personal data from users, which greatly surprised their users. For example, participants were consistently surprised by the fact that a flashlight app needed to know their unique phone ID as well as their precise location.

5.3.3 Expectation, Purpose, and Comfort Level

When participants were surprised by access to a sensitive resource, they also found it difficult to explain why the resource was needed. Note, in the *expectation condition*, participants were only informed about which resources were accessed without information on the purpose of access. This is similar to what the existing Android permission list conveys to users. In this condition, we observed a very strong correlation (r = 0.91) between the percentage of expectations and average comfort ratings. In other words, the **perceived necessity of resource access was directly linked to users' subjective feelings**, which guided the way users made trust decisions on mobile apps.

We also found that, even if users were fully aware of which resources were used, they still **had a difficult time understanding why the resources were needed**. We compared the reasons our participants provided in the expectation condition against the ground truth from our app analysis. In most cases, the majority of participants could not correctly state why a given app requested access to a given resource. When resources were accessed for functionality purposes, participants generally had better answers; however, accuracy never exceeded 80%. When sensitive resources were used for multiple purposes, the accuracy of answers tended to be much lower. Note that these results are for a situation in which participants were paid to read the description carefully. Many of them had even used some of these apps in the past. We believe for general Android users, their ability to guess answers would have been even worse.

Given the lack of clarity as to why resources are accessed, users must deal with significant uncertainties when making trust decisions regarding installing and using a given mobile app. We observed that, for the four types of sensitive resources (i.e., device ID, contact list, network location, and GPS location), **participants, in general, felt more comfortable when they were informed of the purposes of a resource access** (see **Table 3**). The differences between the comfort ratings were statistically significant in paired *t*-tests. For example, concerning accessing the device ID, the average comfort rating in the purpose condition was 0.3 higher than in the expectation condition ($t(55) = 7.42$, $p < 0.0001$). This finding suggests that providing users with reasons why their resources are used not only gives them more information to make better trust decisions, but can also ease concerns caused by uncertainties. Note that informing users about the "purpose" for collecting their information is a common expectation in many legal and regulatory privacy frameworks. Our results confirm the importance of this information. This finding also provides us with a strong rationale to include the purpose(s) of resource access in our new design of privacy summary interface.

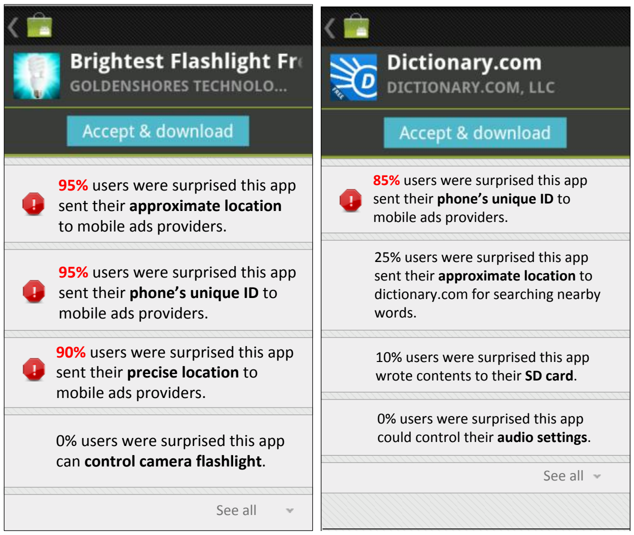
5.4 How to Apply the Results? Preliminary Design of a New Privacy Summary Interface

In the above section, we have discussed how we identify the purpose of sensitive data disclosure in mobile apps and how to capture users' expectation of mobile apps, in the remaining of this chapter we apply these finding in designing better privacy interfaces. The objective is to display richer and more pertinent information to users in a compact and understandable way.

The first design we come up is a text-based design directly inherit the layout and color

Resource Type	comfort rating w/ purpose(std)	comfort rating w/o purpose (std)	df	T	p
Device ID	0.47 (0.30)	-0.10 (0.41)	55	7.42	0.0001
Contact List	0.66 (0.22)	0.16 (0.54)	24	4.47	0.0002
Network Location	0.90 (0.53)	0.65 (0.55)	28	3.14	0.004
GPS Location	0.72 (0.62)	0.35 (0.73)	23	3.60	0.001

Table 7 Comparison of comfort ratings between the expectation condition (2nd column) and the purpose condition (3rd column). Standard deviations are shown between parentheses. When participants were informed of the purpose of resource access, they generally felt more comfortable. The differences were statistically significant for all four types of resources. The comfort ratings were ranging from -2.0 (very uncomfortable to +2.0 (very comfortable).



95% users were surprised this app sent their **approximate location** to mobile ads providers.

95% users were surprised this app sent their **phone's unique ID** to mobile ads providers.

90% users were surprised this app sent their **precise location** to mobile ads providers.

0% users were surprised this app can **control camera flashlight**.

85% users were surprised this app sent their **phone's unique ID** to mobile ads providers.

25% users were surprised this app sent their **approximate location** to dictionary.com for searching nearby words.

10% users were surprised this app wrote contents to their **SD card**.

0% users were surprised this app could control their **audio settings**.

Figure 18: A mockup interface of our newly proposed privacy summary screen, taking the Brightest FlashLight and the Dictionary app as examples. The new interface provides extra information of why certain sensitive resources are needed and how other users feel about the resource usages. Warning sign will appear if more than half of the previous users were surprised about this resource access.

schedule from existing permission screen. This new privacy summary interface features two crucial attributes identified in our previous study, namely *expectation* and *purpose*. In this preliminary design, we directly leverage other users' mental models and highlight their surprises. By presenting the most common misconceptions about an app, we can rectify people's mental models and help them make better trust decisions. We also provide the *purposes of resource access* to give users more explanations in our new summary interface.

Previous research has discussed several problems with the existing Android permission screens [54, 77], including:

- The wording of the permission list contains too much technical jargon for lay users.
- They offer little explanations and insight into the potential privacy risk.
- A long list of permissions makes users experience warning fatigue.

With these problems in mind, in addition to the two identified key features, we proposed several principles for our own design:

- Using simple terms to describe the relevant resources. For example, instead of using "coarse (Network) location", we use the term "approximate location".
- Only displaying the resources that have greater impact on users' privacy, such as location, device ID, storage, contact list etc. Users could choose to check out other low-risk resources by clicking "See all".
- Sorting the list based on users' expectation as captured through crowdsourcing. We order the list so that the more surprising resource usages are shown first.

* p <0.05 ** p<0.005	**# of Mentioning Privacy Concern (out of 20)**		**Accuracy (max=1.0)**			**Time spent (sec)**		
App Name	Existing	Proposed	Existing	Proposed	p	Existing	Proposed	p
Brightest Flashlight	4	6	0.58	0.86	**	74.59	65.11	
Dictionary	1	3	0.73	0.91	**	68.21	43.92	**
Horoscope	3	7	0.75	0.95	*	68.41	48.72	*
Pandora	3	3	0.68	0.94	**	76.86	76.82	
Toss it	4	13	0.61	0.88	**	67.43	57.10	

Table 8. Comparisons between the existing Android permission screen (permission condition) and our newly proposed privacy summary (new interface condition). Our new interface makes users more aware of the privacy implications and is easier to understand. Users in general spent less time on these newly proposed interfaces but got more fine-grained information.

- Highlighting important information. We bold the sensitive resources mentioned in text, and use a warning sign and striking color to highlight the suspicious resource usages, i.e. when the surprise value exceeds a certain threshold.

Figure 18 shows two examples of our new privacy summary interface. To make the comparison more symmetric, our design uses the same background colors and patterns that were used in the Android permission screen at the time of the study. In this study, we used the data collected in our previously described crowdsourcing study to mock up the privacy summary interfaces for five mobile apps, namely Brightest Flashlight Free, Dictionary, Horoscope, Pandora, and Toss it.

We used AMT to conduct a between-subjects user study to evaluate our new privacy summary interface. Participants were randomly assigned to one of two conditions. In one condition, participants were shown the original permission screen that the current Google Play Store uses. In the other condition, participants were shown our new interfaces. We evaluated the new privacy summary interface from three perspectives. The first was *privacy awareness* (i.e., whether users were more aware of the privacy implications). This was measured by counting the number of participants who mentioned privacy concerns when justifying their recommendation decisions. The second was *comprehensibility* (i.e., how well users understood the privacy summary). This was measured by the accuracy in answering questions about app behavior. The third was *efficiency* (i.e., how long it took participants to understand the privacy summary), which was measured by the number of seconds participants spent reading the privacy summary screens.

The comparisons between the two conditions are summarized in Table 8. Generally speaking, participants in the new interface condition weighted their privacy more when they made decisions about whether the app was worth recommending. More people in this condition mentioned privacy-related concerns when they justified their choices. When we asked participants in both conditions to specify the resources used by the target apps, those in the new interface condition demonstrated a significantly higher accuracy compared to their counterparts. Furthermore, except for the Pandora app, participants in the new interface condition, on average, spent less time reading the privacy summaries; however, the time difference was not always statistically significant. This finding suggests that we can provide more useful information without requiring users to spend more time to understand it.

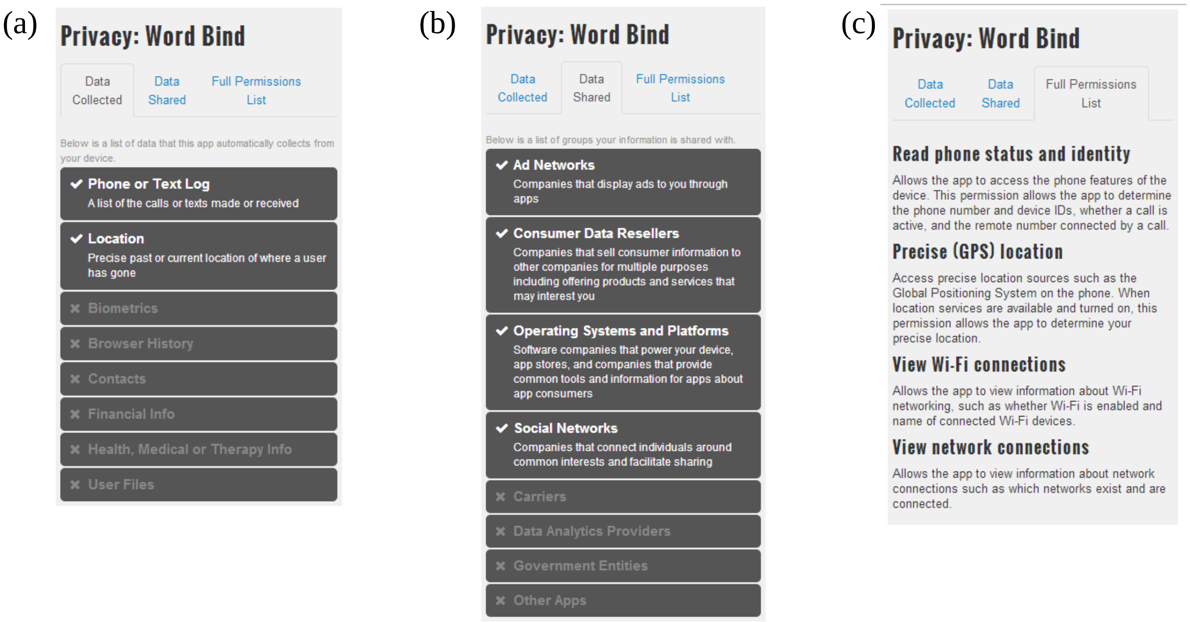
5.5 *Privacy Interfaces With Different Layouts*

Though improved from the existing Android permission screen, the above mentioned privacy interface is still text-based, hence might not be optimal for users to view at a glance. In this sub-section, we present three new designs that build on the expectation and purpose work, using the same crowdsourcing approach, but opting to present the key information in more understandable layouts.

5.5.1 Proposed Privacy Interfaces

Our three designs included a matrix view that shows what permissions an app uses and how those permissions are used (e.g. advertising, analytics, etc), a list view that shows the same information grouped by permissions, and another list view grouped by how permissions are used. As baselines, we compared our designs against Android's interface available since April 2013 (see Figure 2b and 2d), and against a design based on the voluntary Code of Conduct proposed by the National Telecommunications & Information Administration (NTIA) [67, 95, 108], who is an agency of the United States Department of Commerce that serves as the President's principal adviser on telecommunications and information policy.

NTIA's Code of Conduct provides general design guidelines but does not specify a particular standardized design at this point. In our study, we created a baseline privacy notice, based on the Code of Conduct and on published mockups developed by a number of the stakeholders [67]. We also chose to use this version based on its similarities to the



(a) Privacy: Word Bind

Below is a list of data that this app automatically collects from your device.

✓ Phone or Text Log
A list of the calls or texts made or received

✓ Location
Precise past or current location of where a user has gone

✖ Biometrics

✖ Browser History

✖ Contacts

✖ Financial Info

✖ Health, Medical or Therapy Info

✖ User Files

(b) Privacy: Word Bind

Below is a list of groups your information is shared with.

✓ Ad Networks
Companies that display ads to you through apps

✓ Consumer Data Resellers
Companies that sell consumer information to other companies for multiple purposes including offering products and services that may interest you

✓ Operating Systems and Platforms
Software companies that power your device, app stores, and companies that provide common tools and information for apps about app consumers

✓ Social Networks
Companies that connect individuals around common interests and facilitate sharing

✖ Carriers

✖ Data Analytics Providers

✖ Government Entities

✖ Other Apps

(c) Privacy: Word Bind

Read phone status and identity

Allows the app to access the phone features of the device. This permission allows the app to determine the phone number and device IDs, whether a call is active, and the remote number connected by a call.

Precise (GPS) location

Access precise location sources such as the Global Positioning System on the phone. When location services are available and turned on, this permission allows the app to determine your precise location.

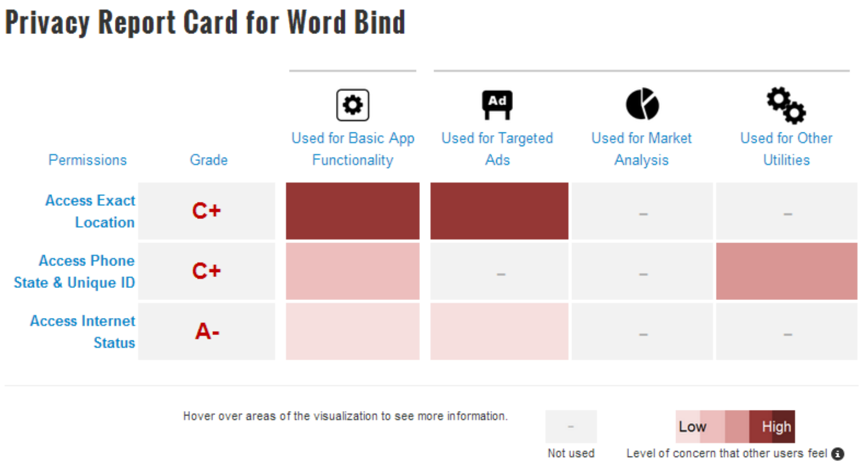
View Wi-Fi connections

Allows the app to view information about Wi-Fi networking, such as whether Wi-Fi is enabled and name of connected Wi-Fi devices.

View network connections

Allows the app to view information about network connections such as which networks exist and are connected.

Figure 19. Implementation of a privacy interface following the NTIA Code of Conduct Guidelines, (a) identifying the types data collected by the app, (b) ways the data may be shared, (c) and a complete listing of the permissions used. We changed the color theme from the proposed mock ups so that they all have the same general texture.



Privacy Report Card for Word Bind

Permissions	Grade	Used for Basic App Functionality	Used for Targeted Ads	Used for Market Analysis	Used for Other Utilities
Access Exact Location	C+	🟥	🟥	–	–
Access Phone State & Unique ID	C+	🟥	–	–	🟥
Access Internet Status	A-	🟥	🟥	–	–

Hover over areas of the visualization to see more information.

– Not used

Low High Level of concern that other users feel

Figure 20. Our matrix interface. Permissions used by the app are on the vertical axis; categories of use are on the horizontal axis. Darker boxes indicate that the behavior was more concerning to other users, based on our estimated crowdsourced data probing people's level of comfort with the app using a given permission for a given behavior. The grades (C+, A-) are based on an average of comfort level across the entire row. Dashes mean that the app does not have a given behavior.

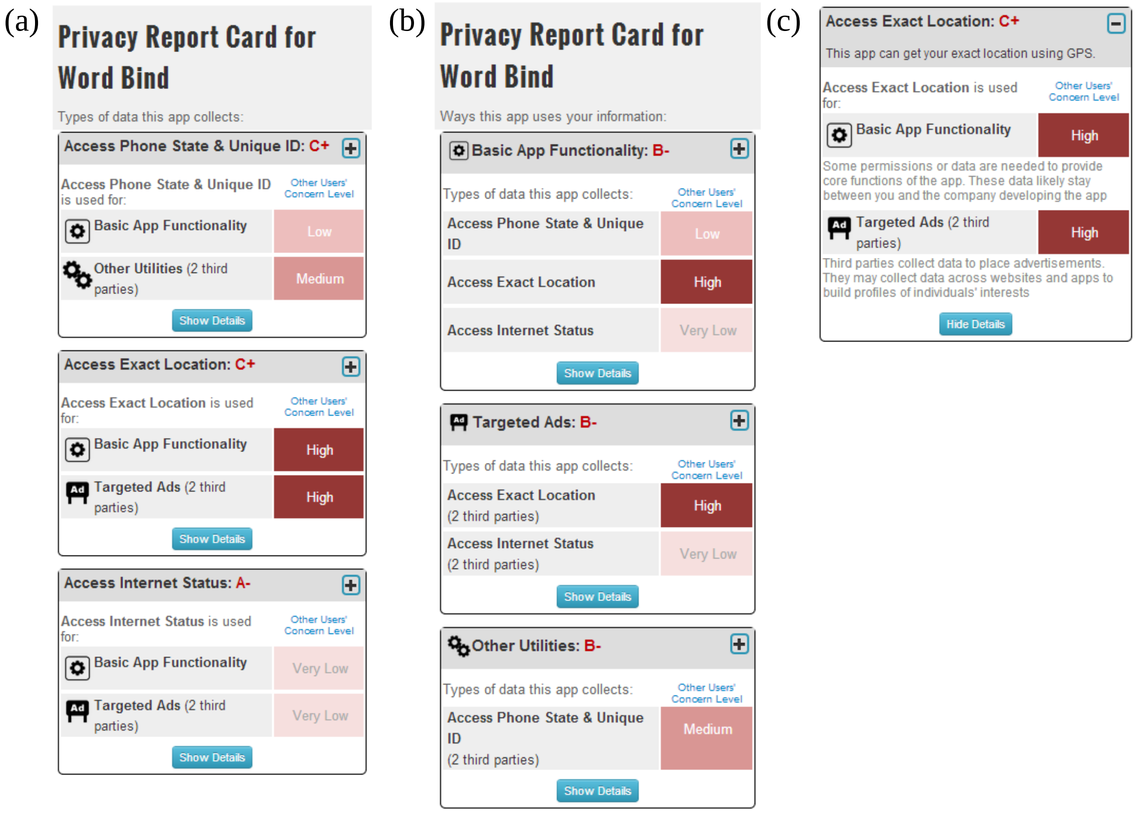
existing Android permission interface. This version of the interface has one tab indicating which data categories are used by the app, a second tab indicating what types of entities may use the data, and a third tab with the full text of which permissions are used (Figure 19). It is primarily text based, and displays the data types that are used first (and grays out the data types that are not).

As we mentioned before, we identified 9 categories of 3rd-party libraries based on the type of services they provide. To simplify our designs, we opted to group the use of permissions into four categories: Basic App Functionality, Advertising, Market Analysis, and Other Utilities. Basic app functionality means that a permission is used by the app and not a third-party. Other utilities include all of the other libraries above other than advertising and market analysis. We grouped these uses together for two reasons. First, advertising and market analysis libraries were found in over 40% of apps that we analyzed, and represented a common piece of functionality across all apps. Second, we wanted to avoid overloading users with too many details, allowing them to focus on information that past work has identified as the most privacy concerning [31, 82, 99].

For our first design, we turned to a matrix visualization to display information about what permissions are used by an app, how the data and information gathered from those permissions are used, and which behaviors users are concerned with (see Figure 20). This design was inspired by past work on privacy nutrition labels [75, 76], which were originally designed for web site privacy policies.

The matrix shows permissions that an app uses along the vertical axis. Categories of how the permissions are used are placed along the horizontal axis (i.e. Basic App Functionality, Targeted Ads, Market Analysis, and Other Utilities). Each box in the grid represents a behavior of the app, corresponding with the permission and type of usage. If the behavior is not exhibited by the app, a gray minus sign is displayed. If the behavior is exhibited by the app, the box is colored red, with darker red colors signifying that other users generally found the behavior concerning, and lighter red colors less concerning. There were a total of five possible gradations. As noted by the privacy nutrition labels work, a matrix approach lets users quickly glance at the visualization and get an idea of the app's behaviors by looking at the total number of shaded cells. Additionally, the concern levels are summarized with a letter grade, from A to F, to provide another way for users to easily skim the visualization. The grades represent a weighted average of concern levels, where A means approximately 80% or more of people do not feel concerned, B approximately 60-80%, and so forth. Grades are also weighted by the sensitivity of the data gathered. Users can hover over the text labels to show short text explanations of the permission as well as the categories of uses.

A readily apparent problem with the matrix is that it does not translate well to a small smartphone screen, in that it requires a great deal of horizontal scrolling. To solve this problem, we created two different list visualizations. For the first list visualization, we created a list grouped by permission types (referred to as List-Permission, see Figure 21a). Essentially, each row of the matrix has its own section, describing how each permission is used. We followed the same conventions used in the matrix, providing a



(a)

Privacy Report Card for Word Bind

Types of data this app collects:

Access Phone State & Unique ID: C+

Access Phone State & Unique ID is used for:	Other Users' Concern Level
Basic App Functionality	Low
Other Utilities (2 third parties)	Medium

Show Details

Access Exact Location: C+

Access Exact Location is used for:	Other Users' Concern Level
Basic App Functionality	High
Targeted Ads (2 third parties)	High

Show Details

Access Internet Status: A-

Access Internet Status is used for:	Other Users' Concern Level
Basic App Functionality	Very Low
Targeted Ads (2 third parties)	Very Low

Show Details

(b)

Privacy Report Card for Word Bind

Ways this app uses your information:

Basic App Functionality: B-

Types of data this app collects:	Other Users' Concern Level
Access Phone State & Unique ID	Low
Access Exact Location	High
Access Internet Status	Very Low

Show Details

Targeted Ads: B-

Types of data this app collects:	Other Users' Concern Level
Access Exact Location (2 third parties)	High
Access Internet Status (2 third parties)	Very Low

Show Details

Other Utilities: B-

Types of data this app collects:	Other Users' Concern Level
Access Phone State & Unique ID (2 third parties)	Medium

Show Details

(c)

Access Exact Location: C+

This app can get your exact location using GPS.

Access Exact Location is used for:	Other Users' Concern Level
Basic App Functionality	High

Some permissions or data are needed to provide core functions of the app. These data likely stay between you and the company developing the app

Targeted Ads (2 third parties)	High

Third parties collect data to place advertisements. They may collect data across websites and apps to build profiles of individuals' interests

Hide Details

Figure 21. Two versions of the list view, optimized for a smartphone screen layout. (a) One list is grouped by permissions, called "List – Permission". For each permission the app uses, the interface displays how that permission is used and a rating of how concerned other users were with that behavior. (b) The second list is grouped by categories of use, called "List – Use". For each use, the interface displays the permissions the app uses as well as a concern rating. (c) Users can click the "show details" button to view longer textual explanations of the permission types and usage categories.

box with a darker shade of red for more concerning actions, though we wrote the level of concern directly over the color (Very Low, Low, Medium, High, Very High), as there is limited screen space to add a legend. A button, labeled *Show Details*, expands the list, allowing users to read more detailed explanations about the types of permissions and categories of data use. The second list visualization groups shows data usage categories (referred to as List-Use, see Figure 21b), by grouping each column of the matrix into its own section, again following the same design conventions as the first list.

5.5.2 Evaluation Methodology

To evaluate our proposed privacy interfaces, we created information for 3 fake apps. We opted for fake apps so that participants would not have any prior experience with the apps. Our apps included Word Bind (a word game app), Friend Pix (a photo and video social networking app), and Alpha Flashlight (a flashlight app). The permissions and behaviors of the fake apps were based on a synthesis of real apps in the Google Play store. We also created concern ratings for each behavior, based on crowd ratings on the behaviors of the synthesized apps. Similar to the previous section, we evaluated these five interfaces with 210 participants from MTurk based on the understandability of these interfaces, the time users spent on reading these interfaces, as well as their impressions of the interfaces. After removing HITs that did not correctly answer a quality control question, a total of 230 HITs were completed. Each of the five interfaces had between 38 and 52 HITs (M= 46). Each participant was presented with the name of the app, user rating, overall privacy grade, app description, and a permission interface. The interface was shown in a narrow frame to mimic a smartphone display, except for the matrix, which was displayed in a wider frame.

Participants were asked questions in three sections. The first section had 6 multiple choice questions, and was designed to see how well the interfaces conveyed information to users. The first question asked what the purpose of the app is, to make sure participants had read all the information. This was also used to help filter participants who did not fully complete the task. The second question asked about a type of permission used by the app. The third and fourth questions asked about why or how permissions are used. The fifth and sixth questions asked about the concern levels regarding certain behaviors. Timing data was collected to see how long it took participants to answer questions in the first section.

The second section was designed to gauge participants' reactions to the user interfaces. Participants were asked to rate on a 5-point Likert scale how concerned they felt about each permission presented in the interface. They were also asked to use 5-point Likert scales to rate how comfortable they would feel downloading the app, how useful they thought the interface was, how difficult they thought it was to understand the information in the interface. Optional open-ended comments were also collected in this section.

The third section consisted of demographics information, including the participant's age, occupation, sex, type of smartphone owned (if any), length of time they owned a smartphone, and approximately how many apps they have installed. Additionally, participants were asked a series of six questions that referred as the simplified Westin's privacy scale in order to determine what Westin's privacy categories they belongs to [80].

Interface	*n*	Total Score	Permissions	How Info is Used	Concern
Google Play	47	3.85	0.89	N/A	N/A
NTIA	38	2.95	0.63	1.21	N/A
Matrix	45	3.60	0.87	1.44	1.28
List-Permission	48	3.54	0.85	1.52	1.17
List-Use	52	3.47	0.85	1.44	1.06
Maximum Possible Score or SubScore:		5	1	2	2

Table 9. Summary of the mean number of questions answered correctly. The total number of questions is listed in the bottom row. Higher scores mean more questions answered correctly.

5.5.3 High Level Results

At a high level, people answered fewer questions correctly with the NTIA interface than the other conditions. However, while participants were asked five multiple choice questions in addition to the quality control questions, the questions were not the same across all conditions, because the interfaces convey different dimensions of information. This information is summarized in Table 9. All five interfaces conveyed information about the permissions an app uses, and all were asked one question about this type of information. This data is treated as binary (the user either answered correctly or incorrectly), and a Pearson Chi-Square test with X^2 *(1, N = 230) = 12.42, p = .014*, suggests that the NTIA interface that we created performed significantly worse at conveying information about what permissions are used.

Participants were asked two questions about how an app might use their data and what information is gathered from permissions, for NTIA, Matrix, List-Permission, and List-Use interfaces. These questions let us compare the understandability of these interfaces. While the participants with the NTIA interface answered fewer questions correctly, a one way ANOVA shows no significant difference between the four conditions, *F(3, 179) = 1.426, p = .237, r = .15*.

Participants were also asked two questions about how concerned other users felt about the app’s permission behaviors, for the Matrix, List-Permission, and List-Use interfaces. These questions let us compare the understandability of crowd comfort levels across these interfaces. While the participants with the Matrix interface on average answered more questions correctly, a one way ANOVA shows no significant difference between these three conditions, *F(2, 142) = 1.107, p = .334., r = .12.*

In short, the understandability test results suggested that the NTIA interface that we created performed significantly worse at conveying information about what permissions are used. At the same time, our new proposed three interfaces had similar understandability comparing to the existing Android permission interface, though providing much richer information.

We also measured how long it took participants to answer the multiple choice information questions (Table 10). There are some outliers creating a large difference in

Interface	***n***	**Mean Seconds**	**Std. Dev. Seconds**	**Median Seconds**	**Mean Rank**
Google Play	47	146.11	67.64	131	83.03
NTIA	38	309.18	386.29	204	131.78
Matrix	45	181.84	124.31	163	110.61
List-Permission	48	207.23	169.31	199	122.83
List-Use	52	235.25	197.05	191.5	130.41

Table 10. Summary of the mean seconds for users to complete the multiple choice section of our proposed interfaces, the standard deviation of seconds, the median number of seconds, and the mean rank.

standard deviations and the distribution of seconds is not normal even without outliers. To adjust for this, we analyzed the data using ranks by performing a Kruskal-Wallis test and found a significant result, H(4) = 16.90, p = .002.

This suggests that the length of time it took for participants to complete the Google Play multiple choice questions was significantly less than in the other interface conditions,. This is further supported in pairwise comparisons, which show that the number of seconds it took in the Google Play condition is significantly shorter than the NTIA condition (p = .008), the List-Permission condition (p = .036), and the List-Use condition (p = .004).

Because the questions were slightly different for the Google Play and Matrix conditions but the same for the Matrix and both List conditions, we perform a Kruskal-Wallis test on the timing data for the three proposed interfaces. We find that the difference is not significant, H(2) = 2.420, p=.298. In other words, **given the similar amount of information, different layouts did not take users significantly longer or shorter time to digest.** Meanwhile, the fact that the NTIA interfaces took the longest time for users to understand also demonstrate the weakness of text-based interfaces.

In addition, we gathered qualitative responses from our participants, which helped us better understand the evaluation results. In our quantitative data analysis, we noticed that the NTIA interfaces were consistently worse than others both in terms of the understandability and the time it cost participants to read. Some reasons were suggested by our participants. For example, one participant commented that there was "*too much text, info hidden in multiple tabs*," which made information hard to find. We emphasize that we only tested one permission interface based on the NTIA Code of Conduct guidelines, and it is possible that others may fare better. Furthermore, most of the mockups presented in [67] use multiple tabs to display information, which may make the NTIA interface harder to understand. More generally, **the lack of matching the specific data types to a specific way the data is used is a general weakness of the NTIA's Code of Conduct guidelines**. One participant wrote that "*I honestly had to use reasoning* *to answer why the app would collect my location…because that figure above doesn't explicitly provide that information*." A revised guideline requiring matching the specific data types to the ways data are used, or new interface that displays this information while still following the current guidelines may help users better understand the information.

There were noticeable differences in these comments between conditions, which confirmed that **users do need to know why their data are used**. In the Google Play and NTIA interface conditions, people usually commented by asking why a permission is being used, such as *"why does it need access to my call log and GPS location," "I'm just curious about why it needs access to my contacts,"* or *"I wonder why a flashlight app would really need all of those permissions."* This makes sense, because the Google Play and NTIA interfaces do not provide specific information about how each permission is used. In contrast, those who cite specific permissions as major concerns in the Matrix and two List conditions are able to cite specific reasons why they do not feel comfortable, saying *"I don't want my location tracked especially when it could be used by third parties for ads," "it should just need your general location for ads,"* and *"I don't like location targeted ads or ads involving my contacts."* Interestingly, the Matrix and List interfaces were also used to help provide positive explanations and make some participants feel more comfortable about the app. There were several positive comments citing specific app behaviors were found in each of these conditions. Participants said *"I can sort of see why it might want your location for ads,"* and *"Many apps use the phones location for ad presentation. This is accepted practice."* This finding is in line with past work [82], which found that offering explanations improved comfort levels.

Furthermore, amongst our three newly proposed interfaces, several participants wrote that they found it difficult to understand the information at first. For example, participants wrote *"it took me a minute to figure it out but once I got it, it was easy,"*, *"it was somewhat difficult to initially understand what it was saying"*. However, while users may have felt that these interfaces were harder to understand at first, there was no evidence in our data. The participants in these conditions did not have significantly different accuracies in answering factual questions, and did not take significantly longer time to complete tasks. So, our results suggest that while the interface did not harm performance, the types of information presented in the interface were new and unfamiliar, and **with repeated use of seeing these interfaces, users should feel better about using the interface.** These findings also suggest there will be challenges in introducing users to new dimensions of mobile privacy information.

5.6 Summary

In this chapter, we introduced a new methodology for disclosing mobile apps' behavioral information to end users. Key contributions include:

- We identified two key features--- expectation and purpose--- that can provide richer information for users to make better privacy decisions. Our approach helps uncover gaps in the user's mental models. We show that these gaps or misconceptions can help inform the design of more effective privacy decision interfaces.

- We demonstrated the feasibility of using crowdsourcing to capture people’s mental models of an app’s privacy-related behaviors in a scalable manner.
- We proposed and evaluated 4 interfaces that make use of the identified key information. These interfaces and their evaluation also shed some light on how additional design elements such as UI layout, colors, order in which information is presented can effect users’ understanding of what an app does and also impact their level of comfort.

6 CHOICES & CONTROLS: HOW TO HELP USERS WITH THEIR PRIVACY SETTINGS?

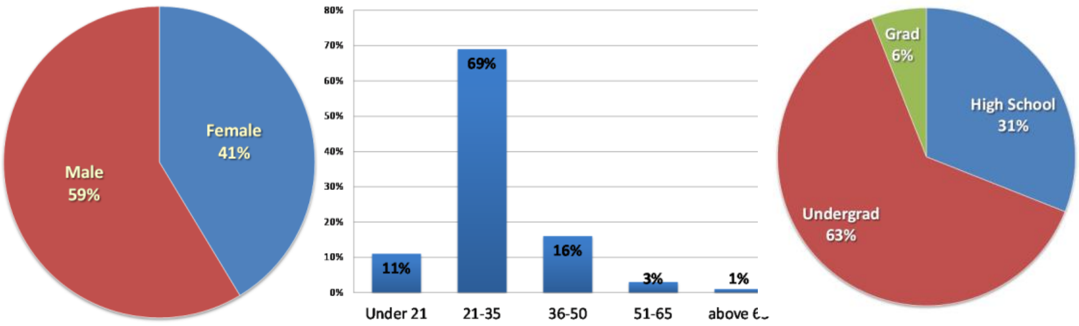
In the previous chapter, we presented the design and evaluation of several alternative user interfaces for informing users about an app's privacy-related behaviors in better ways. In this chapter, we focus our discussion on modeling users' mobile app privacy preferences in order to provide users assistance in the process of configuring privacy settings.

6.1 Usability Challenges in Managing Privacy Settings

Several early mobile app privacy solutions have involved allowing users to control individual app permission. For example, TISSA [123] and MockDroid [29] allow users to substitute fake information in response to API calls by an app. A similar approach has also been made available to users of jailbroken iPhones through an app known as "Protect My Privacy" [13, 19]. Most recently both iOS and Android have moved to such an approach. For instance, in iOS6, users have the ability to selectively toggle permissions on and off for individual apps, with these permissions including access to one's location, calendar, photos, reminders and more. In June 2013, with the introduction of Android 4.3, Android introduced similar settings in the context of a hidden app permission manager referred to as 'App Ops". These developments can be viewed as a direct reflection of the diverse privacy preferences revealed through our own research, as reported in the previous Chapter. While users express concerns about many permissions, they do not all feel the same way, hence the need to provide them with the ability to decide for themselves whether or not they want to grant a particular permission to a particular app.

While in theory the fine-grained permission interfaces that have emerged over the past few years empower users to control their permission settings, they also make unrealistic assumptions about a user's ability and willingness to control such a large number of settings. According to a report from Nielsen in 2012 [87], the average number of apps installed by a smartphone user was around 41 in the year of 2012. Given that on average 4 permissions are required by an app, an average smartphone user would have to make over a hundred privacy decisions to configure the permissions settings of all these apps. For more active users, the number of decisions they need to make might be well over a thousand. Furthermore, users might not completely understand the privacy implications behind their decisions. In short, providing users with the ability to control their data is not sufficient. To make the privacy settings usable and practical, there is an urgent need for trusted tools that can guide users through the configuration process and reduce the number of privacy decisions a user actually has to make.

Quantitatively modeling users' mobile app privacy preferences is the first step we take to get closer to this goal. To address the usability issue, we leverage user-oriented machine learning techniques to identify a set of representative privacy profiles that users can choose as their default privacy settings. We present our techniques and key findings in the following sections.



(a) Gender

(b) Age

(c) Education

Figure 22: Demographic information of Amazon Mturk workers who participated in our data collection.

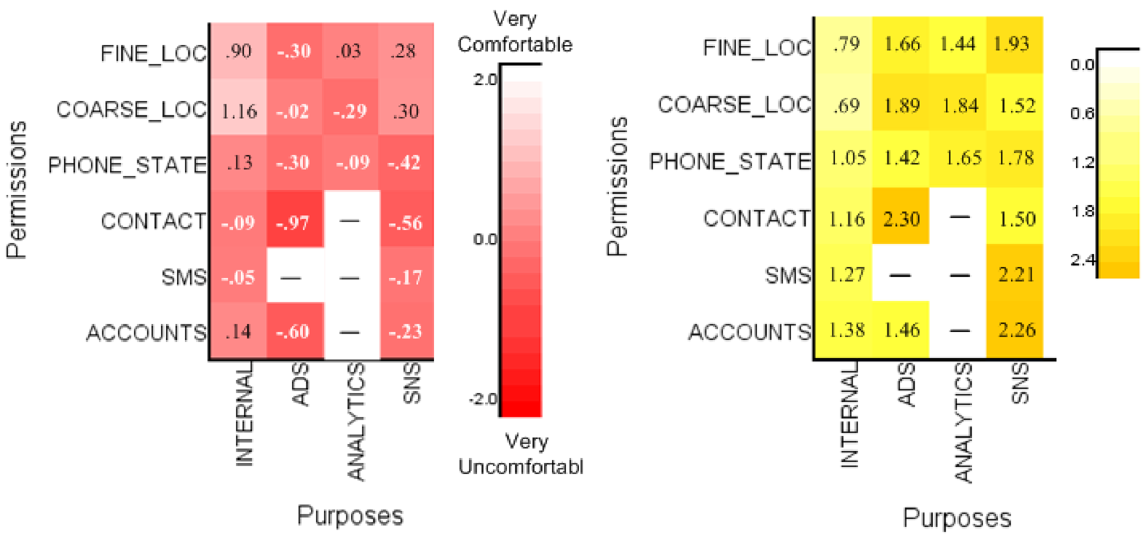
6.2 Crowdsourcing Study

To gather enough user preference data for machine learning analysis, we used Amazon Mechanical Turk to conduct a study similar to the one previously described in Chapter 5.3. Participants were shown the app's icon, screen shots, and description. Participants were asked if they expected this app to access certain type of private information and were also asked how comfortable they felt allowing this app to access their information for the given purposes. The permission usage and the purpose were based on the static analysis discussed in Chapter 3. We also collected demographic information of our participants including gender, age and education background to help us analyze our data. As in our previous crowdsourcing study, we restricted our participants to U.S. smartphone users with previous HIT approval rate higher than 90%.

We scaled up our study to over 1200 HITs, probing 837 mobile apps that we randomly sampled from the top 5000 most popular free apps. Each HIT examined one app-permission- purpose triple. For example, in one HIT, participants were asked to express their level of comfort in letting Angry Birds (app) access their precise location (ACCESS_FINE_LOCATION permission) for delivering targeted ads (purpose). The data collection ran for 3 weeks starting on June 15th, 2013. After the data collection period, we first eliminated responses that failed the qualification questions[15], and then we eliminated 39 HITs because they had less than 15 responses. This yielded a dataset of 21,657 responses contributed by 725 MTurk workers.

We did not specifically control the gender ratio or any other demographic composition of our participants. As shown in Figure 22, among these participants, 41% of them were female; 69% of participants were between 21 and 35, 16% of them are between 36 and 50 (see Figure 22 (b)). We also observed that more than 60% of the participants were reported to have a bachelor's degree or equivalent and 6% had a master's degree or PhD. The average education level of our participants was significantly higher than the average

[15] In the qualification questions, we asked participants to choose the appropriate category of the apps to test if they read the app description carefully.



(a) Average user preferences

(b) Variances in user preferences

Figure 23: (a) The average self-reported comfort ratings of different permission usages. The blue indicates more comfortable, and the magenta indicates more concerning. (b) The variances in users ratings. For most cases, there are significant variances among users in their privacy preferences.

education level of the entire U.S. population as reported in [28]. Compared to the demographics of crowdworkers as reported in [104], our participant pool contains more people with bachelor's degrees and fewer with graduate degrees.

This difference in demographics may be due to self-selection, since usually crowdworkers would be more likely to work on HITs that interest them. However, other data collection methods, such as Internet surveys, often have similar sampling problems. While this sample bias has to be taken into account when interpreting our results, we suspect that our study is no worse than most others in terms of the representativeness of our participant pool.

6.3 Users' Average Preferences and Their Variances

To visualize our results, we aggregate self-reported comfort ratings by permission and purpose. Figure 23 (a) shows the average preferences of all 725 participants, where white indicates participants were very comfortable (2.0) with the disclosure, and red indicates very uncomfortable (-2.0). In other words, cells that are in the darker shades of red indicate a higher level of concern.

The three use cases with the highest levels of comfort were: (1) apps using location information for their internal functionality (fine loc: *M=0.90*, coarse loc: *M=1.16*); (2) SNS libraries bundled in mobile apps using users' location information so this context information can be used in sharing (fine loc: *M=0.28*, coarse loc:*M=0.30*); (3) apps accessing smartphone states, including unique phone ID, and account information for internal functionality (*M=0.13*).

For the remaining cases, users expressed different levels of concerns. Users were generally uneasy with (1) targeted advertising libraries accessing their private information, especially for their the contact list (*M=-0.97*), and accounts information that store on their mobile devices (*M=-0.60*); (2) SNS libraries that access their unique phone ID (*M=-0.42*), contact list (*M=-0.56*) as well as information related to their communication and web activities such as SMS (*M=-0.17*) and accounts information (*M=-0.23*); (3) mobile analytic libraries accessing their information such as location (*M=-0.29*) as well as phone states (*M=-0.09*).

This aggregation of data gave us a good starting point to spot general trends in users' privacy preferences. At the same time, these are averages and, as such, they do not tell us much about the diversity of opinions people might have. In our previous research of users' location privacy preferences, an important lesson we learned is that users' privacy preferences are very diverse. Therefore, we plot the variances of user preferences of the same use cases in Figure 23 (b) to see the variance of people's preferences. In this figure, the darker shades of yellow indicate higher variance among users' comfort rating for different purposes.

Figure 23 (b) shows that users' preferences are definitely not unified. Variances are larger than 0.6 (of a rating in a [-2,+2] scale) in all cases. In 25% of cases, variances exceed 1.8. Users' disagreements were highest in the following cases, including:

(1) SNS libraries accessing users' SMS information as well as their accounts[16]
(2) targeted advertising libraries accessing users' contact list.
(3) users' location information being accessed by all kinds of external libraries.

This high variance in users' privacy preferences suggests that having a single one-size-fits-all privacy setting for everyone would not work well – at least for those settings with a high variance. We cannot simply average the crowdsourced user preferences and use them as default settings as suggested in [19]. This begs the question of whether users could possibly be subdivided into a small number of groups or clusters of like-minded users for which such default settings (different settings in different groups) could be identified.

6.4 Clustering on Users' Preferences

Given the variances identified above, a natural solution is to see if there are large groups of people with similar preferences. In order to identify these groups, we need to properly encode each user's preferences into a vector and trim the dataset to prevent over-fitting. More specifically, we conducted three kinds of preprocessing before feeding the dataset into various clustering algorithms. First, we eliminated participants who contributed less than 5 responses to our data collection, since it would be different to categorize participants if we know too little about their preferences. This step yields a total number of 479 unique participants with 20,825 responses. On average, each participant

[16] In fact, SNS libraries usually use GET_ACCOUNTS permission in the process of authenticating users. Users had extreme responses for this use case mainly due to that they have limited knowledge of the authentication process.

	Connectivity		Dunn Index		Silhouette Width	
Rank	Dist- aggl-k	Value	Dist- aggl-k	Value	Dist- aggl-k	Value
Top 1	CBR-CRT-3	8.64	CBR-AVG-4	0.60	CBR-AVG-4	0.55
Top 2	CBR-AVG-4	8.78	MHT-AVG-4	0.55	CBR-CTD-4	0.54
Top 3	ECL-AVG-4	11.23	CBR-CTD-5	0.53	CBR-WAR-4	0.42

Table 11: Top 3 clustering configurations for each internal measure. Clusters obtained by using Canberra distance and average linkage method with k=4 (CBR-AVG-4) ranks first in Dunn Index (should be maximized) and Silhouette Width (should be maximized) and ranks second in the connectivity (should be minimized). We select this configuration as it has the best performance overall.

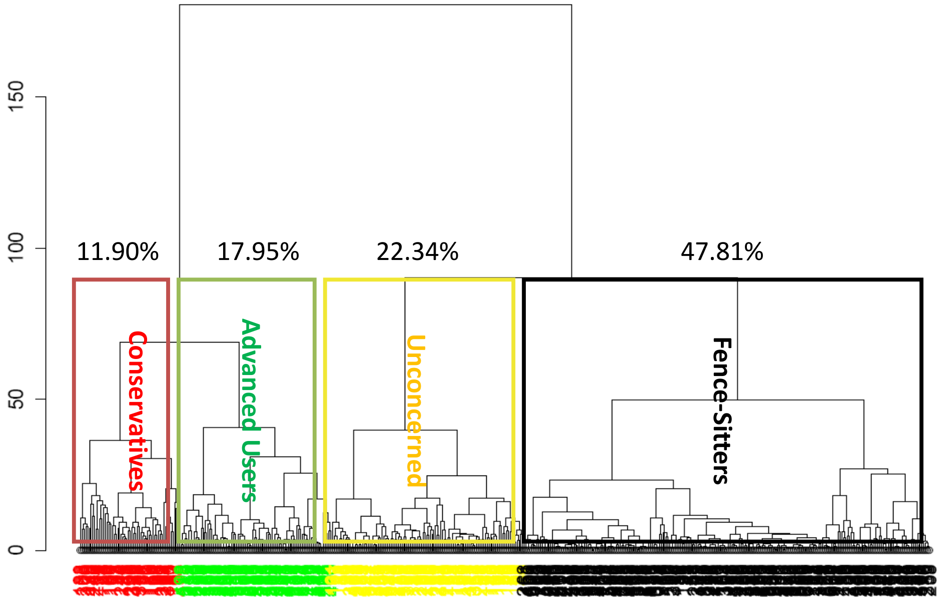


Figure 24: The resulting dendrogram produced by hierarchical clustering with Canberra distance and average linkage agglomerative method. Four different colors are used to indicate the cluster composition when *k=4*. We also overlay the cluster names on the dendrogram which will be explained in Chapter 6.4.2.

contributed 43.5 responses *(SD=38.2, Median=52)*. Second, we aggregated a participant's preferences by averaging their indicated comfort levels of letting apps use specific permissions for specific purposes. "NA" is used if a participant did not have a chance to indicate his/her preferences for a give app-permission-purpose triple. Lastly, for each missing features ("NA"), we find the *k (k=10)* nearest neighbors which have that feature. We then impute the missing value by using the imputation function on the k values from the neighbors.

After these preprocessing steps, we obtain a matrix of 77 columns and 479 rows, where each row of the matrix represents a participant. This preference matrix is free of missing values.

6.4.1 Algorithms and Clustering Results

Just like in Section 4.2 and 4.3, because our primary objective is to identify easily interpretable clusters with semantically meaningful boundaries, we opt again for hierarchical clustering techniques. By comparing the ranking of all configurations and the *k* value (see Table 11), we obtain the best clusters by using Canberra distance and average linkage method with *k=4*.

Figure 24 illustrates the resulting dendrogram produced by the above mentioned clustering configurations, where four different colors indicate the four clusters when *k=4*. Among the four identified clusters, the largest one (colored in black in Figure 24) includes 47.81% of instances, whereas the smallest cluster (colored in red) includes 11.90% instances.

We interpret the clustering results and discuss the characteristics of each of the four clusters in the following sub-section.

6.4.2 Making Sense of Privacy Profiles

To make sense of what these clusters mean, we compute the centroid of each cluster by averaging the features of all the instances within the same cluster. Note that the previously imputed missing values are excluded in computing the centroids. We call these centroids "privacy profiles", since they represent the average privacy preferences of each group of users. We then use a heat map to visualize these privacy profiles[17] in Figure 25. The vertical dimension of these heat maps represents the uses of different permissions, and the horizontal dimension represents why certain permission is used. We use two colors to indicate people's preferences. Color white indicates that participants feel comfortable with a certain type of disclosure where as the red indicates the level of uncomfortable. The grid with a short dash means we do not have data for this grid. We also assign each privacy profile with a name that highlights its characteristics to help distinguish these clusters.

The (Privacy) Conservatives: Although conservatives form the smallest group among the four clusters, they still take up 11.90% of our participants (top-left in Figure 25). Among all four heat maps, this privacy profile has the largest area covered in red (feeling uncomfortable). In general, this type of users feels very uncomfortable letting their sensitive personal information (such as location and unique phone ID) be used by external libraries. They also feel uncomfortable if mobile apps uses their unique phone ID, contact list or SMS internally if the necessity of using these sensitive personal data is not visible to them.

The Unconcerned: This group of participants take up 23.34% of the all the participants and forms the second largest cluster in our dataset (top-right in Figure 25). The heat map of this privacy profile has the largest area covered in light color (feeling comfortable). In general, users who share this privacy profile feel comfortable disclosing their sensitive personal data (almost) in every case, no matter who is collecting their data and for what

[17] In these heat map visualizations, we only display the most important six permissions and four purposes which strongly differentiate these privacy profiles.

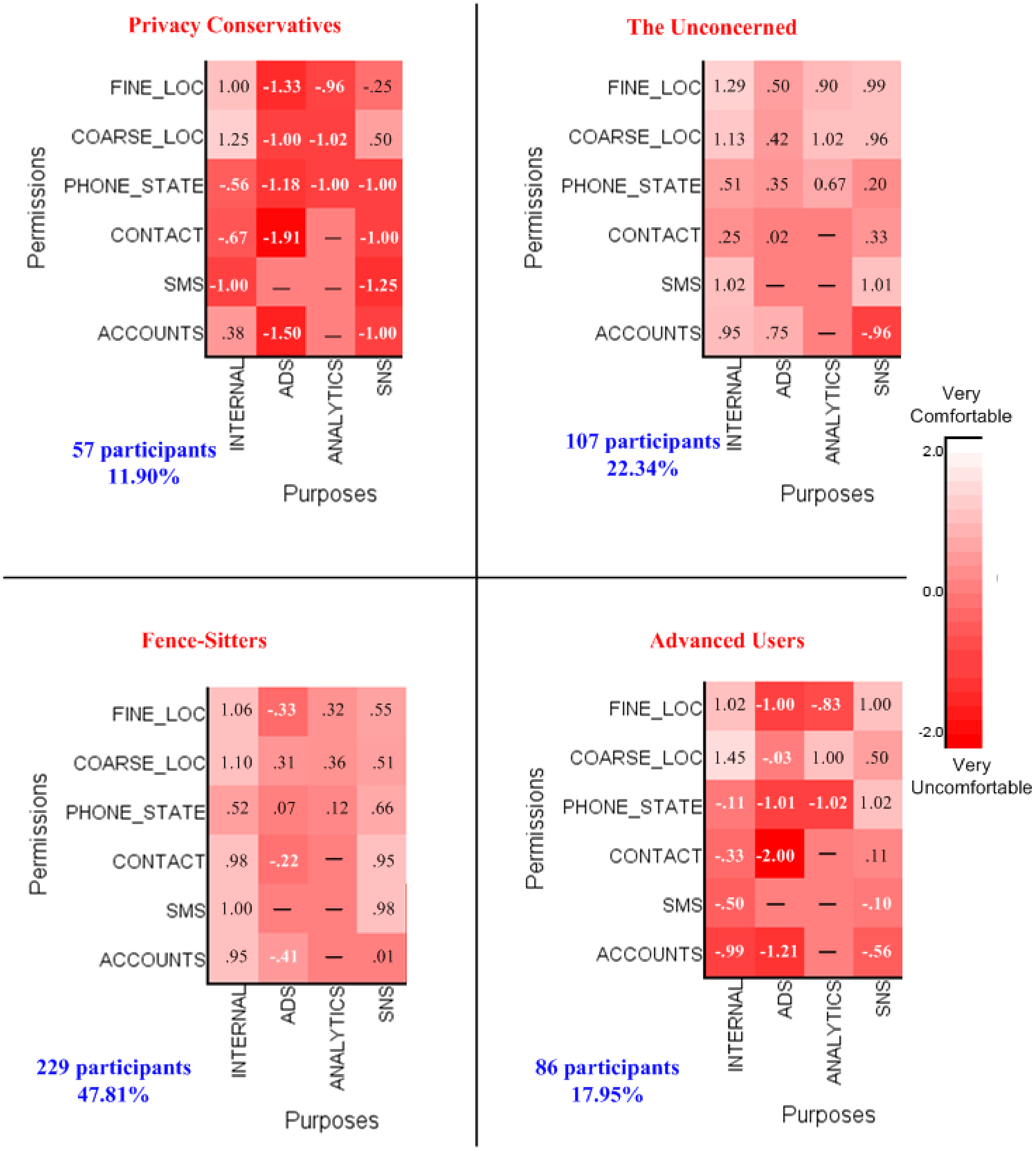


Figure 25: Four privacy profiles identified through clustering. We assign each cluster a

purposes. The only concerning (red) grid in the heat map is regarding to the cases where SNS libraries access GET_ACCOUNTS information. We suspect this outstanding grid is caused by lack of data, or alternatively there might be a large portion of users do not understand the meaning of this permission.

The Fence-Sitters: We named this privacy profile "The Fence-Sitters" because most participants within this cluster do not have strong opinions for a large portion of the use cases (bottom-left in Figure 25). As the largest cluster we identified, this group takes up nearly 50% of the population. Unsurprisingly, this group of participants feels very comfortable letting mobile apps access their sensitive personal data for internal functionality purposes. With regard to the cases where their information is consumed by 3rd-party libraries such as for delivering targeted ads or conducting mobile analytics,

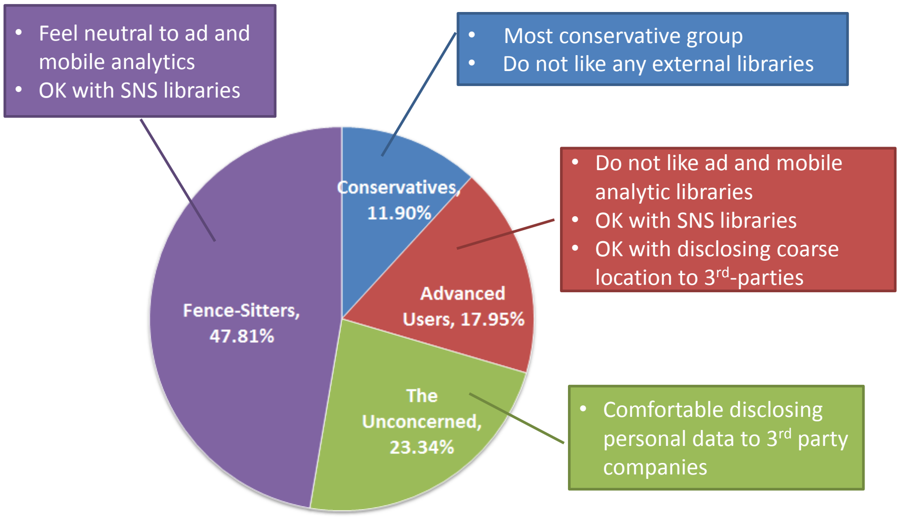


Figure 26 . Summary of the four identified user clusters.

they expressed attitudes very close to neutral (i.e. neither comfortable nor uncomfortable). This characteristic is rather visible on the heat map that large portions of the grids are in pink (close to the middle color in the legend). This group of participants also feels OK disclosing all types of sensitive personal data to SNS libraries consistently. Without further investigation, it is hard to know exactly why so many users belong to this category. We suspect that to some level it might be caused habituation (or warning fatigue) that a significant portion of users have already formed the habit of clicking through the permission screen as warned by Felt. et al. in [50].

The Advanced Users: The advanced user group is 17.95% of the population (bottom-right in Figure 25). The reason we named this group as “advanced users” is because these users appears to have a more nuanced understanding of sensitive data usages. In general, most of them feel comfortable with their sensitive data being used for internal functionality and by SNS libraries. We believe they feel okay with the latter scenario because they still have control over the disclosures, since these SNS libraries often let people confirm sharing before transmitting data to corresponding social network sites. In addition, this group of users dislikes targeted ads and mobile analytic libraries, but still feels generally agreeable in disclosing context information in a lower granularity (i.e. coarse location). This observation again suggests that this group of users has more insights than others in expressing their privacy preferences.

Figure 26 summarizes the outstanding characteristics of each cluster and shows the portions they take up in the participant pool. By identifying these four major privacy profiles, we have a clearer sense of how different users view various sensitive data usage patterns. In the following sections, we will talk about how these privacy profiles can be applied to benefit multiple stakeholders.

6.5 Implications of Privacy Profiles

Gender	**Conservatives**	**Unconcerned**	**Advanced**	**Fence-Sitter**	**Total**
Female (0)	21	42	25	101	189
Male (1)	36	65	61	128	290
Total	57	107	86	229	479

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Conservatives	57	36	0.63158	0.23684
Unconcerned	107	65	0.60748	0.2407
Advanced	86	61	0.7093	0.20862
Fence-Sitter	229	128	0.55895	0.24761

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	1.462004	3	0.487335	2.049186	0.106169	2.623677
Within Groups	112.9639	475	0.237819			
Total	114.4259	478				

Table 12. Gender distributions of each user cluster and ANOVA analysis results. We see no statistically significant differences among the gender distribution of these groups.

Age Group	**Conservatives**	**Unconcerned**	**Advanced**	**Fence-Sitter**	**Total**
< 21 (1)	11	39	12	17	79
21-35 (2)	41	62	59	170	332
36-50 (3)	4	6	13	30	53
51-65 (4)	1	0	2	7	10
> 65 (5)	0	0	0	5	5
Total	57	107	86	229	479

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Conservatives	57	107	1.877193	0.252506
Unconcerned	107	181	1.691589	0.328513
Advanced	86	176	2.046512	0.374282
Fence-Sitter	229	434	1.895197	0.585459

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	6.222892	3	2.074297	4.598546	0.003482	2.623677
Within Groups	214.2615	475	0.451077			
Total	220.4843	478				

Table 13. Age distribution of each user group and ANOVA analysis. The unconcerned group on average is significantly younger, and the advanced users are on average significantly older than the other groups combined.

In this section, we discuss how the identified four privacy profiles can be used to assist users in configuring their privacy settings. Ideally, if we can identify which cluster a user

Education	**Conservatives**	**Unconcerned**	**Advanced**	**Fence-Sitter**	**Total**
High School (1)	20	39	17	67	143
Bachelor's (2)	37	64	56	147	304
Master's or higher (3)	0	4	13	15	32
Total	57	107	86	229	479

SUMMARY

Groups	*Count*	*Sum*	*Average*	*Variance*
Conservatives	57	94	1.649123	0.23183
Unconcerned	107	179	1.672897	0.297655
Advanced	86	173	2.011628	0.364569
Fence-Sitter	229	406	1.772926	0.299088

ANOVA

Source of Variation	*SS*	*df*	*MS*	*F*	*P-value*	*F crit*
Between Groups	6.828428	3	2.276143	7.523031	6.30E-05	2.623677
Within Groups	143.7144	475	0.302557			
Total	150.5428	478				

Table 14. Distribution of education level of each user group and ANOVA analysis. The conservatives and the unconcerned are both with lower education levels comparing to the remaining groups combine, whereas the advanced users are more likely with higher level of education.

belongs to without having them explicitly label privacy preferences for individual apps and permissions, an appropriate privacy profile can be directly applied to users' privacy settings as a default. The question is, how can we select which privacy profile is closest to the user's true privacy preferences? We discuss this issue in the following sub-sections.

6.5.1 Privacy profiles and demographic information

The first approach we tried is to see if users' demographic information --- including gender, age and education level --- are possibly sufficient to predict their privacy preferences. To test this hypothesis, we summarize the distribution of gender, age and education level of each user cluster and perform analysis of variance (ANOVA) to see if there are significant differences in these distributions.

Table 12- Table 14 presents the results from the ANOVA test. In general, we found that in regard to the gender distribution, a one-way analysis of variance yield NO significant differences between groups, *F(3, 475)=2.049, p=0.106*. The detailed mean and variance values can be found in Table 12.

For age distribution, we encoded the age groups as (1:= under 21, 2:= age 21-35, 3:=age 36-50, 4:=age 51-65, 5:=above 65) in our calculation. A one-way analysis of variance reveals significant differences between groups in regard to age distribution, *F(3, 475)=4.598, p=0.003*. Post hoc analyses also reveals that the unconcerned group on average are younger *(M=1.69, SD=0.57)* than other groups combined *(M=1.91, SD=0.76)*, and the advanced user group on average are older *(M=2.05, SD=0.61)* than other groups combined *(M=1.83, SD=0.71)*. The mean and variance of each group are shown in Table 13.

We also performed a similar test on the education level of all four groups of participants. We encoded the education levels such that "1" stands for high school or lower level of education, "2" stands for bachelor or equivalent level of degrees, and "3" stands for master's or higher level of degrees. An ANOVA test shows that the effect of education was strongly significant, *F(3, 475)=7.52, p=6.3E-05* . Post hoc analyses show that the conservatives *(M=1.65, SD=0.48)* and the unconcerned *(M=1.67, SD=0.54)* have lower education levels compared to the remaining groups combined *(M=1.85, SD=0.57)*, and the advanced users *(M=2.01, SD=0.60)* are more likely to have a higher level of education.

Although there are statistically significant effects in demographics, a regression from demographic information to the cluster label yields accuracy no better than directly putting every user as Fence-Sitters. In other words, we should not directly use gender, age, or education level to infer which privacy profile should be applied to individual user. This does not mean however that in combination with other factors, these attributes would not be useful. Below, we seek more deterministic methods to assign privacy profiles in the following sub-section.

6.5.2 Identifying privacy profiles: what questions to ask

In Android 4.3, users are given the ability to fine-tune their privacy preferences by turning on and off permission usages of individual app through "App Ops" [114]; however, as we discussed before, the usability issues hinder the ability of lay users to make use of these controls. It would be extremely handy if, when a user boots up her Android device for the first time (or possibly at a later time), the operating system could ask her a small number of questions to determine which privacy profile is the best match for her. The profile could then be used to select default privacy settings for her. As she downloads apps on her smartphone, "App Ops" or some equivalent functionality would then be able to automatically infer good default settings for her.

Similar ideas have been suggested for helping users set up their location sharing rules [103] [92]. In particular Wilson et al. in [119] describe a simple wizard for the Locaccino system, where a small number of questions were asked to guide users through the selection of good default location sharing profiles. In this section, we suggest that a similar method could be used to identify a small number of questions and help identify good mobile app privacy profiles for individual users.

Given the four privacy profiles that we identified in Chapter 6.4, we have several observations that to some extent can differentiate different groups of users:

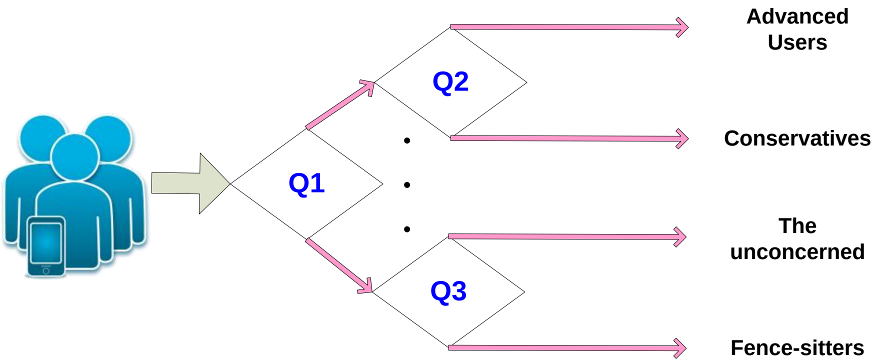


Figure 27 . Users are asked a set of general questions to determine which cluster they belong to.

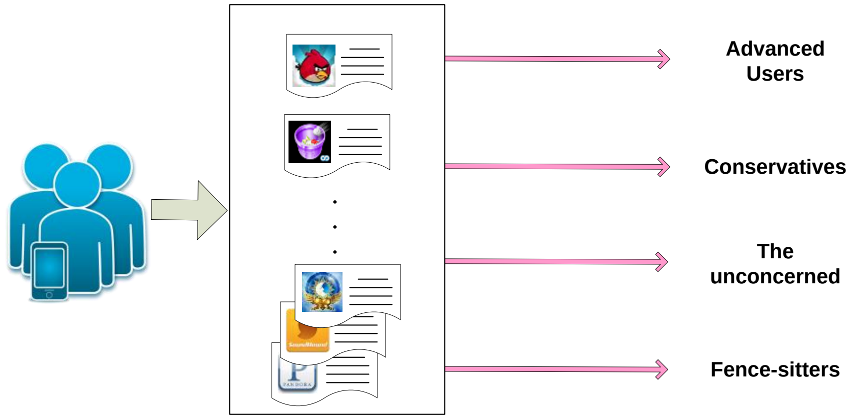


Figure 28 . Users are asked to rate a set of apps to determine which cluster they belongs to.

- **Observation 1: Regarding Advertisement**
  With respect to the cases where mobile apps share users' data with advertising agencies, users in general have three attitudes. Privacy conservatives and advanced users are very uncomfortable with this use of their information, whereas the unconcerned, while not completely comfortable with it, generally find it acceptable. For fence-sitters, attitudes are much less negative than for conservatives but still slightly on the negative side.

- **Observation 2: Regarding Mobile Analytics and Coarse Location**
  We observe similar attitudes for mobile analytics, except that the conservatives and the advanced users can be further distinguished by their preferences when it comes to letting mobile analytics libraries collect their coarse location information.

- **Observation 3: Regarding SNS and Fine Location**
  With respect to the cases where libraries for social network sites access users' fine location through mobile apps, the privacy conservatives stand out since they are the only group of users that generally express negative comfort values for this usage. Advanced users are similar to the unconcerned in this case. These two groups feel comfortable letting SNS libraries access their fine location, though they might have

different reasons. The unconcerned feel comfortable are more likely because they do not care whether their data flow to SNSs, whereas the advanced users are more likely because they know that SNSs still need their explicit actions (e.g. pressing the "Share" button) to make the data disclosure happen.

Similarly, usage patterns with regard to contact list, SMS and phone status can also to some level help differentiate between users in different clusters. We can leverage these observations by centering our questions around these findings. As illustrated in Figure 27, the ideal scenario is that based on their answers to these questions, users are accurately assigned to the most appropriate clusters. For example, we can ask one questions with regard to targeted advertising, such as *"How do you feel letting mobile apps access your personal data for delivering targeted ads?"* or questions about mobile analytics, such as *"How do you feel letting mobile apps sending your approximate location for market analysis purpose?"* Of course, the exact wording and expressions used in these questions require thorough user studies to verify.

Alternatively, instead of asking users these abstract questions, we could present users with a small set of example apps together with the description of their privacy-related behaviors. Users could rate each app based on its sensitive data usages, and we could then classify users based on these ratings, as illustrated in Figure 28. This would work particularly well if we could identify a small number of particularly popular apps that are sufficient to differentiate between users - say just asking people whether they feel comfortable sharing with their location with Angry Bird for advertising purpose and whether they feel comfortable sharing their location on Facebook through the app Scope. Again, how to select the most effective set of apps and how many apps should be included in this process are open questions that will warrant further investigating.

In spite of the uncertainty, we are able to quantitatively demonstrate the theoretical improvement we can achieve in estimating users' true preferences by using privacy profiles. We compute how accurately we can predict users' privacy preferences in three settings as follows:

(1) **Baseline setting (baseline):**
In the baseline setting, we take the average preferences over all users as the only privacy profile users can choose from. This is similar to recommendations currently made by ProtectMyPrivacy [13]. Then we compute the mean square error (MSE) between the average profile and users true preferences (i.e. the comfort rating they contributed for various conditions).

(2) **One question setting (Q1):**
In this setting, we are aware of the four privacy profiles identified previously. However, we assume that only one question can be asked to determine which privacy profile should be chosen. We further assume that this question can accurately distinguish the conservatives and the advanced users apart from the remaining users, thus separating users into two groups. MSE is calculated to evaluate the estimation.

(3) **Two question setting (Q1...Qn):**

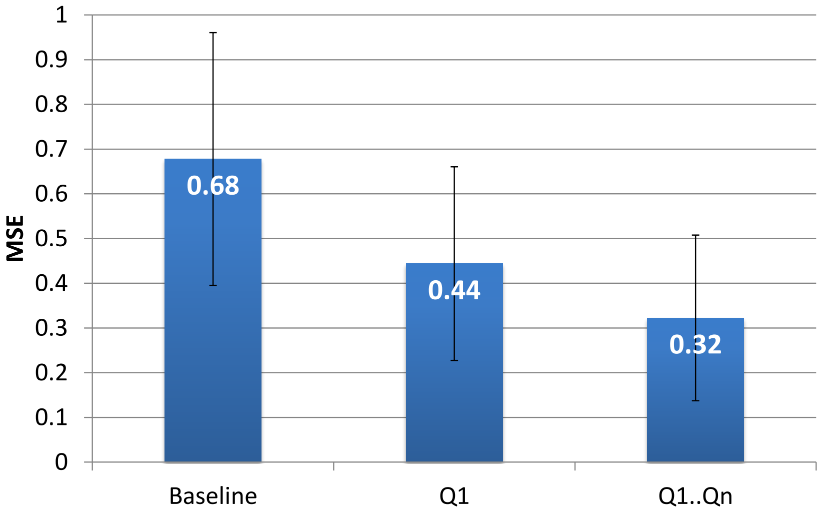


Figure 29: Privacy preference estimation in three settings. Vertical bars indicate the standard deviations. In the baseline setting, the grand average preferences are used as the only privacy profile; in Q1 setting, only one question is allowed to ask to determine the appropriate privacy profile; in Q1...Qn setting, we assume a perfect set of questions are asked. The MSEs of the latter two conditions give theoretical upper bound of the best performance potentially can be achieve if proper questions are chosen.

> In this setting, we assume that we have a set of questions that can accurately separate and assign users to the most appropriate privacy profiles. We calculate the MSE between the chosen privacy profile and users real preferences to evaluate the estimation.

Figure 29 illustrates the average and standard deviation of MSEs in these three conditions. In the baseline condition, the MSE is as high as 0.68, where adding one question to determine the privacy profiles reduces the average MSE by 33.8%. Further adding more questions can reduce the average MSE in the baseline setting by 52.9%. We also observe that the standard deviation of MSE in Q1 and Q1...Qn conditions are significantly lower than the one in the baseline. Note we should emphasize again that these average MSEs only provide a theoretical bound of the best we can achieve in estimating users real preferences by using privacy profiles. This is a big assumption that the questions can perfectly differentiate users and select the optimal privacy profiles for them. In reality, this assumption needs to be tested through thorough user studies.

In addition to choosing appropriate privacy profile as a starting point, users' later interactions with their privacy settings could be used as input into reinforcement learning algorithms to refine models of a user's particular privacy preferences as suggested in [39, 79, 92].

6.5.3 Other potential applications

In addition to serving as default privacy settings, we believe that the identified privacy profiles can also be applied to other domains.

For instance, the privacy profile information could be integrated into an app recommender system. Existing app recommender systems, including the ones provided by Google Play, usually give recommendations only based on users' interests in terms of what functionalities apps provide. This leaves users to filter out apps that they have privacy concerns with among all these recommended apps. By knowing what users' privacy preferences are like, app recommender systems can take the privacy dimension into consideration, providing recommendation that are also based on whether apps' behaviors align with users' privacy preferences.

The identified privacy profiles, as well as their approximate proportions of the user population, can also provide important information to app developers in making better design decisions. App developers can quantitatively estimate the proportion of users who might not install their app, or the proportion of users who might turn off certain permission if this app bundles with certain 3rd-party libraries. In this way, developers are able to make more grounded choices with regard to the trade-off between user experiences and profit. For example, if a developer plans to include a targeted ads library that aggressively collects users' contact list, he might consider the fact that the inclusion of this library might turn away over 70% of users due to privacy considerations; thus he should further evaluate if it is worth bundling this library.

In short, the findings that we present in this chapter provide important lessons about mobile app users, and also point out a way to make privacy settings potentially usable to end users. There is still plenty effort can be made on each step of modeling users' privacy preferences. We are also aware that users' privacy preferences might keep on evolving and are influenced by the introduction of new technologies and the habituation effect that formed through interacting with the same practices for a long time. Therefore, in addition to all the techniques we proposed, proper user education on mobile app privacy is still crucial and needs to be promoted in the long run.

7 CONCLUSION AND FUTURE WORK

In this chapter, we summarize the contributions of this thesis and point out some directions for future work.

7.1 Thesis Summary

The main purpose of this thesis work is to complement existing mobile app privacy research by providing important knowledge on the end-use's side and bridge the gap between security-oriented approach and the user research in mobile app privacy. More specifically, this thesis made significant contributions in the following aspects.

Firstly, this thesis presents the results of a large scale static analysis of over 100,000 smartphone apps across the entire Google Play store, providing detailed results regarding the typical usage patterns of mobile apps in consuming users' sensitive data. We specifically focused on analyzing what 3rd-party libraries were bundled in apps, since the inclusion of 3rd-party libraries provides us some of the semantics of how sensitive data are used. This analysis also produces a valuable dataset that other researchers can use to dig deeper in the apps' behavior analysis.

Secondly, this thesis contribute to the design of privacy interfaces by identifying two key features that can be help users make better privacy decisions. They are the "purpose" which refers to the reason why users' sensitive data are used and "expectation" which refers to whether certain app's behavior breaks users' expectation. We operationalize privacy by capturing people's expectations as well as reflecting other's expectations directly in a privacy summary to emphasize places where an app's behavior surprises its users. We propose a series of user interfaces that visualize these features in different layouts. Evaluation results show that our new proposed new interfaces can greatly arise privacy awareness and are well-received by users.

Last but not the least, we utilize crowdsourcing to collect the mobile app privacy preferences of over 700 users with regard to over 875 apps. Based on the collected data, four distinct privacy profiles are identified, providing reasonable default settings for users to choose from, which significantly mitigate the usability problem suffered by permission level privacy configurations.

Although this thesis focused mainly on the free Android apps in Google Play, we believe that the models we built based on users' privacy preferences and the identified segments of users may also potentially be applied to other App Markets such as Amazon App Store or Apple App Store. We expect that the knowledge we discovered on apps and the lessons we learned in informing users of privacy-related information as well as managing users' privacy settings can also help market owners to improve their current privacy frameworks. Especially for Android 4.3 and onward, in which users are able to manage permission uses for individual apps, the operating system could naturally crowdsource users' privacy preferences as one type of Google usage data that users can opt to contribute. A significant portion of the methodologies discussed in this thesis can be directly applied to these crowdsourced data to build models of mobile users in the wild.

Meanwhile, we acknowledge that privacy has many facets. This thesis only points out some possible ways to address this problem. We believe other facets, such as educating users and app developers, improving and enforcing laws and regulations, are also crucial for protecting mobile users' privacy.

7.2 Future Work

This thesis work also leaves several directions worth improving and extending.

7.2.1 Leverage NLP techniques to further understand the functionality of the app

In this thesis, we only focus on the privacy aspects of mobile app. This because extracting high-level functionality features of apps through static analysis is challenging and the categories provided by Google are simply too vague and inaccurate to infer the functionality of apps. In Chapter 3.4, we demonstrated that by leveraging NLP techniques on user reviews, we can identify the functional defects or performance issues of mobile apps. Similarly, by applying NLP techniques on apps' descriptions and user reviews, we believe more attributes can be generated to depict apps' functionalities. These features can greatly enrich the dataset resulted from app analysis, hence providing more facets for understanding and categorizing mobile apps.

7.2.2 User studies to evaluate identified privacy profiles

A series of user studies can complement this thesis in two ways. First, we want to see if the privacy profiles we identified with crowd workers (in Chapter 6.4) are representative enough to describe users in the wild. Second, we want to identify the optimal set of questions (mentioned in Chapter 6.5.2) that can accurately guide users to the appropriate privacy profile, which might require multiple iterations on the question sets by using both qualitative and quantitative evaluation methods.

7.2.3 Design, implement and deploy a privacy wizard

From a more systems-oriented point of view, future work can also focus on building a privacy wizard with the identified questions as a module in the smartphone privacy framework to reinforce the privacy profiles we identified. This privacy wizard should also have the ability to gradually refine users' privacy settings based on their interactions with privacy settings. Deploying this wizard to real users in field studies can also as part of the design process to evaluate the usability of this privacy wizard. It would also be interesting to see how such a privacy wizard influences users' privacy preferences or their interaction pattern with mobile apps and smartphones in general.

8 REFERENCES

[1]"Amazon's Mechanical Turk." Available: www.mturk.com

[2]"Androguard." Available: http://code.google.com/p/androguard/

[3]"apkinspector." Available: http://code.google.com/p/apkinspector/

[4]"APKProtect." Available: http://www.apkprotect.com/

[5]"BrightKite." Available: http://brightkite.com

[6]"DED: Decompiling Android Applications."

[7]"Dexter." Available: http://dexter.dexlabs.org/

[8]"Facebook Places ". Available: http://www.facebook.com/places/

[9]"foursquare." Available: http://foursquare.com/

[10]"Google Latitude." Available: http://www.google.com/latitude

[11]"Locaccino: A User-Controllable Location-Sharing Tool." Available: http://www.locaccino.org/

[12]"Loopt." Available: http://loopt.com

[13]"Protect My Privacy: iOS App Privacy Protection." Available: http://www.protectmyprivacy.org/

[14]"Standford Topic Modeling Toolbox." Available: http://nlp.stanford.edu/software/tmt/tmt-0.4/.

[15]"Katz v United States 389 U.S. 347." Available: http://en.wikipedia.org/wiki/Katz_v._United_States

[16]"Neer." Available: http://www.neerlife.com/

[17]"android-apktool." Available: http://code.google.com/p/android-apktool/

[18]"App Profiles." Available: https://play.google.com/store/apps/details?id=com.appdescriber&feature=search_result#?t=W251bGwsMSwxLDEsImNvbS5hcHBkZXNjcmliZXIiXQ..

[19]Y. Agarwal and M. Hall, "ProtectMyPrivacy: detecting and mitigating privacy leaks on iOS devices using crowdsourcing," In Proc. *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, 2013.

[20]R. Amadeo. "App Ops: Android 4.3's Hidden App Permission Manager, Control Permissions for Individual Apps! ." Available: http://www.androidpolice.com/2013/07/25/app-ops-android-4-3s-hidden-app-permission-manager-control-permissions-for-individual-apps/

[21]AppBrain. "Most popular Android market categories." Available: http://www.appbrain.com/stats/android-market-app-categories

[22]R. Balebako, J. Jung, W. Lu, L. F. Cranor, and Carolyn Nguyen, ""Little Brothers Watching You:" Raising Awareness of Data Leaks on Smartphones," In Proc. *SOUPS*, 2013.

[23]R. Balebako, R. Shay, and L. F. Cranor, "Is Your Inseam a Biometric? Evaluating the Understandability of Mobile Privacy Notice Categories," 2013.

[24]D. Barrera, H. G. Kayacik, P. C. v. Oorschot, and A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to android," In Proc. *CCS*, 2010.

[25]Z. Benenson, F. Gassmann, and L. Reinfelder, "Android and iOS users' differences concerning security and privacy," In Proc. *CHI '13 Extended Abstracts*, 2013.

[26]M. Benisch, P. Kelley, N. Sadeh, and L. Cranor, "Capturing location-privacy preferences: quantifying accuracy and user-burden tradeoffs," *Personal and Ubiquitous Computing*, 2010.

[27]M. Benisch, N. Sadeh, and T. Sandholm, "Methodology for designing reasonably expressive mechanisms with application to ad auctions," In Proc. *Proceedings of the 21st international jont conference on Artifical intelligence*, 2009.

[28]U. S. C. Bereau. "Educational Attainment." Available: http://www.census.gov/hhes/socdemo/education/index.html

[29]A. Beresford, A. Rice, and N. Sohan, "MockDroid: trading privacy for application functionality on smartphones," In Proc. *HotMobile*, 2011.

[30]M. S. Bernstein, G. Little, R. C. Miller, B. Hartmann, M. S. Ackerman, D. R. Karger, D. Crowell, and K. Panovich, "Soylent: a word processor with a crowd inside," In Proc. *UIST*, 2010.

[31]T. Book, A. Pridgen, and D. S. Wallach, "Longitudinal Analysis of Android Ad Library Permissions," *CoRR*, vol. abs/1303.0857, 2013.

[32]B. Brown, A. Taylor, S. Izadi, A. Sellen, J. Kaye, and R. Eardley, "Locating Family Values: A Field Trial of the Whereabouts Clock," In Proc. *UbiComp*, 2007.

[33]I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: behavior-based malware detection system for Android," In Proc. *SPSM*, 2011.

[34]Y. Chen, H. Xu, Y. Zhou, and S. Zhu, "Is this app safe for children?: a comparison study of maturity ratings on Android and iOS applications," In Proc. *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[35]J. Cheng. "Pandora sends user GPS, sex, birthdate, other data to ad servers." Available: http://arstechnica.com/gadgets/news/2011/04/pandora-transmits-gps-gender-birthdate-other-data-to-ad-servers.ars

[36]E. Chin, A. P. Felt, K. Greenwood, and D. Wagner, "Analyzing inter-application communication in Android," In Proc. *MobiSys*, 2011.

[37]E. Chin, A. P. Felt, V. Sekar, and D. Wagner, "Measuring User Confidence in Smartphone Security and Privacy," In Proc. *Soups*, 2012.

[38]E. K. Choe, J. Jung, B. Lee, and K. Fisher, "Nudging People Away From Privacy-Invasive Mobile Apps Through Visual Framing," In Proc. *Interact*, 2013.

[39]J. Cranshaw, J. Mugan, and N. Sadeh, "User-Controllable Learning of Location Privacy Policies with Gaussian Mixture Models," In Proc. *AAAI*, 2011.

[40]J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh, "Bridging the gap between physical location and online social networks," In Proc. *UbiComp*, 2010.

[41]J. S. Downs, M. B. Holbrook, S. Sheng, and L. F. Cranor, "Are your participants gaming the system?: screening mechanical turk workers," In Proc. *Proceedings of the 28th international conference on Human factors in computing systems*, 2010.

[42]J. C. Dunn, "Well separated clusters and optimal fuzzy-partitions," *Journal of Cybernetics*, vol. 4, pp. 95-104, 1974.

[43]M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "PiOS: Detecting Privacy Leaks in iOS Applications," In Proc. *NDSS*, 2011.

[44]S. Egelman, A. P. Felt, and D. Wagner, "Choice Architecture and Smartphone Privacy: There's a Price for That," In Proc. *WEIS*, 2012.

[45]W. Enck, "Defending Users against Smartphone Apps: Techniques and Future Directions," in *LNCS*. vol. 7093, ed, 2011.

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

[47]W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, "A Study of Android Application Security," In Proc. *USENIX Security Symposium*, 2011.

[48]W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," In Proc. *CCS*, 2009.

[49]A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," In Proc. *CCS*, 2011.

[50]A. P. Felt, S. Egelman, M. Finifter, D. Akhawe, and D. Wagner, "How to Ask for Permission," In Proc. *HotSec*, 2012.

[51]A. P. Felt, S. Egelman, and D. Wagner, "I've Got 99 Problems, But Vibration Ain't One: A Survey of Smartphone Users' Concerns," UCB/EECS-2012-70, 2012.

[52]A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner, "A survey of mobile malware in the wild," In Proc. *SPSM*, 2011.

[53]A. P. Felt, K. Greenwood, and D. Wagner, "The effectiveness of application permissions," In Proc. *USENIX conference on Web application development*, 2011.

[54]A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android Permissions: User Attention, Comprehension, and Behavior," In Proc. *Soups*, 2012.

[55]A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, and E. Chin, "Permission re-delegation: attacks and defenses," In Proc. *USENIX conference on Security*, 2011.

[56]M. Frank, D. Ben, A. P. Felt, and D. Song, "Mining Permission Request Patterns from Android and Facebook Applications," In Proc. *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, 2012, pp. 870-875.

[57]M. Franklin, D. Kossmann, T. Kraska, S. Ramesh, and R. Xin, "CrowdDB: Answering Queries with Crowdsourcing," In Proc. *SIGMOD*, 2011.

[58]FTC. "Fair Information Practice Principles." Available: http://www.ftc.gov/reports/privacy3/fairinfo.shtm

[59]B. Fu, J. Lin, L. Li, C. Faloutsos, J. Hong, and N. Sadeh, "Why People Hate Your App—Making Sense of User Feedback in a Mobile App Store," In Proc. *KDD*, 2013.

[60]F. Gandon and N. Sadeh, "Semantic web technologies to reconcile privacy and context awareness," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 1, pp. 241-260, 2004.

[61]N. Good, R. Dhamija, J. Grossklags, D. Thaw, S. Aronowitz, D. Mulligan, and J. Konstan, "Stopping spyware at the gate: a user study of privacy, notice and spyware," In Proc. *SOUPS*, 2005.

[62]Google. "Jelly Bean--- Android 4.3 APIs." Available: http://developer.android.com/about/versions/jelly-bean.html

[63]S. Grobart. "The Facebook Scare That Wasn't." Available: http://gadgetwise.blogs.nytimes.com/2011/08/10/the-facebook-scare-that-wasnt/

[64]M. Hamblen. "Smartphone shipments to hit 1B -- up 40% -- in 2013." Available: http://www.computerworld.com/s/article/9242100/Smartphone_shipments_to_hit_1B_up_40_in_2013

[65]J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics,* vol. 21, pp. 3201-3212, 2005.

[66]P. Heymann and H. Garcia-Molina, "Turkalytics: analytics for human computation," In Proc. *Proceedings of the 20th international conference on World wide web*, 2011.

[67]HFEDERMAN. "NTIA User Interface Mockups." Available: http://www.applicationprivacy.org/2013/07/25/ntia-user-interface-mockups/

[68]P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droids you're looking for: retrofitting android to protect data from imperious applications," In Proc. *CCS*, 2011.

[69]S. Huang, F. Proulx, and C. Ratti, "iFIND: a Peer-to-Peer Application for Real-time Location Monitoring on the MIT Campus," In Proc. *CUPUM*, 2007.

[70]G. Iachello, I. Smith, S. Consolvo, G. D. Abowd, J. Hughes, J. Howard, F. Potter, J. Scott, T. Sohn, J. Hightower, and A. Lamarca, "Control, Deception, and Communication: Evaluating the Deployment of a Location-Enhanced," ed: unknown, 2008.

[71]P. G. Ipeirotis, F. Provost, and J. Wang, "Quality management on Amazon Mechanical Turk," In Proc. *Proceedings of the ACM SIGKDD Workshop on Human Computation*, 2010.

[72]C. Jensen and C. Potts, "Privacy policies as decision-making tools: an evaluation of online privacy notices," In Proc. *CHI*, 2004.

[73]J. Jeon, K. K. Micinski, J. A. Vaughan, N. Reddy, Y. Zhu, J. S. Foster, and T. Millstein, "Dr. Android and Mr. Hide: Fine-grained security policies on unmodified Android," 2012.

[74]J. Jung, S. Han, and D. Wetherall, "Short paper: enhancing mobile application permissions with runtime feedback and constraints," In Proc. *SPSM*, 2012.

[75]P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder, "A "nutrition label" for privacy," In Proc. *SOUPS*, 2009.

[76]P. G. Kelley, L. Cesca, J. Bresee, and L. F. Cranor, "Standardizing privacy notices: an online study of the nutrition label approach," In Proc. *CHI*, 2010.

[77]P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A Conundrum of permissions: Installing Applications on an Android Smartphone," In Proc. *USEC*, 2012.

[78]P. G. Kelley, L. F. Cranor, and N. Sadeh, "Privacy as Part of the App Decision-Making Process (CMU-CyLab-13-003)," Carnegie Mellon University2013.

[79]P. G. Kelley, P. H. Drielsma, N. Sadeh, and L. F. Cranor, "User-controllable learning of security and privacy policies," In Proc. *Proceedings of the 1st ACM workshop on Workshop on AISec*, 2008.

[80]P. Kumaragura and L. F. Cranor, "Privacy Indexes: A Surey of Westin's Studies," Carneigie Mellon University CMU-ISRI-05-138, 2005.

[81]Liato. "Python Android Market Library." Available: https://github.com/liato/android-market-api-py

[82]J. Lin, S. Amini, J. Hong, N. Sadeh, J. Lindqvist, and Joy Zhang, "Expectation and Purpose: Understanding Users' Mental Models of Mobile App Privacy through Crowdsourcing," In Proc. *Ubicomp'12*, 2012.

[83]J. Lin, M. Benisch, N. Sadeh, J. Niu, J. Hong, B. Lu, and S. Guo, "A comparative study of location-sharing privacy preferences in the United States and China," *Personal and Ubiquitous Computing*, pp. 1-15, 2012/10/01 2012.

[84]J. Lin, G. Xiang, J. I. Hong, and N. Sadeh, "Modeling people's place naming preferences in location sharing," In Proc. *UbiComp*, 2010.

[85]G. Little, L. B. Chilton, M. Goldman, and R. C. Miller, "TurKit: human computation algorithms on mechanical turk," In Proc. *Proceedings of the 23nd annual ACM symposium on User interface software and technology*, 2010.

[86]G. Liu, G. Xiang, B. A. Pendleton, J. I. Hong, and W. Liu, "Smartening the crowds: computational techniques for improving human verification to fight phishing scams," In Proc. *SOUPS*, 2011.

[87]I. Lunden. "U.S. Consumers Avg App Downloads Up 28% To 41; 4 Of 5 Most Popular Belong To Google." Available: http://techcrunch.com/2012/05/16/nielsen-u-s-consumers-app-downloads-up-28-to-41-4-of-the-5-most-popular-still-belong-to-google/

[88]T. W. Malone, R. Laubacher, and C. N. Dellarocas, "Harnessing Crowds: Mapping the Genome of Collective Intelligence," *SSRN eLibrary*, 2009.

[89]C. D. Manning, P. Raghavan, and H. Schutze, "Hierarchical Clustering," in *Introduction to Information Retrieval*, ed: Cambridge University Press, 2008.

[90]W. Mason and S. Suri, "Conducting Behavioral Research on Amazon's Mechanical Turk," *Behavior Research Methods*, *Forthcoming*, 2010.

[91]A. M. McDonald and L. F. Cranor, "The Cost of Reading Privacy Policies," *I/S: A Journal of Law and Policy for the Information Society*, 2008.

[92]J. Mugan, T. Sharma, and N. Sadeh, "Understandable Learning of Privacy Preferences Through Default Personas and Suggestions," Carnegie Mellon University CMU-ISR-11-112,2012.

[93]M. Nauman, S. Khan, and X. Zhang, "Apex: extending Android permission model and enforcement with user-defined runtime constraints," In Proc. *ASIACCS*, 2010.

[94]D. Norman, *The design of everyday things*: Basic Books, 2002.

[95]NTIA. "Privacy Multistakeholder Process: Mobile Application Transparency." Available: http://www.ntia.doc.gov/other-publication/2013/privacy-multistakeholder-process-mobile-application-transparency

[96]J. Oberheide. "Dissecting Android Bouncer." Available: http://jon.oberheide.org/files/summercon12-bouncer.pdf

[97]D. Octeau, W. Enck, and P. McDaniel, "The ded Decompiler. Technical Report NAS-TR-0140-2010, ," 2010.

[98]S. Patil and J. Lai, "Who Gets to Know What When: Configuring Privacy Permissions in an Awareness Application," In Proc. *CHI*, 2005.

[99]P. Pearce, A. P. Felt, G. Nunez, and D. Wagner, "AdDroid: privilege separation for applications and advertisers in Android," In Proc. *ASIACCS*, 2012.

[100]T. Pedersen, "Unsupervised Corpus-Based Methods for WSD," in *Word Sense Disambiguation*. vol. 33, E. Agirre and P. Edmonds, Eds., ed: Springer Netherlands, 2006, pp. 133-166.

[101]R. "clValid: Validation of Clustering Results." Available: http://cran.r-project.org/web/packages/clValid/index.html

[102]R. "Hierarchical Cluster Analaysis." Available: http://stat.ethz.ch/R-manual/R-patched/library/stats/html/hclust.html

[103]R. Ravichandran, M. Benisch, P. G. Kelley, and N. Sadeh, "Capturing Social Networking Privacy Preferences. Can Default Policies Help Alleviate Tradeoffs

between Expresiveness and User Burden?," In Proc. *the Privacy Enhancing Technologies Symposium*, 2009.

[104]J. Ross, L. Irani, M. S. Silberman, A. Zaldivar, and B. Tomlinson, "Who are the crowdworkers?: shifting demographics in mechanical turk," In Proc. *CHI '10 Extended Abstracts on Human Factors in Computing Systems*, 2010.

[105]P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics,* vol. 20, pp. 53-65, 1987.

[106]J. M. Rzeszotarski and A. Kittur, "Instrumenting the crowd: using implicit behavioral measures to predict task performance," In Proc. *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011.

[107]N. Sadeh, J. Hong, L. Cranor, I. Fette, P. Kelley, M. Prabaker, and J. Rao, "Understanding and Capturing People's Privacy Policies in a Mobile Social Networking Application," *The Journal of Personal and Ubiquitous Computing*, 2009.

[108]N. Singer. "Under Code, Apps Would Disclose Collection of Data." Available: http://www.nytimes.com/2013/07/26/technology/under-code-apps-would-disclose-collection-of-data.html?=_r=6&_r=1&

[109]D. J. Solove, "A Taxonomy of Privacy," *University of Pennsylvania Law Review, Vol. 154, No. 3, January 2006*.

[110]K. P. Tang, J. Lin, J. I. Hong, D. P. Siewiorek, and N. Sadeh, "Rethinking location sharing: exploring the implications of social-driven vs. purpose-driven location sharing," In Proc. *UbiComp*, 2010.

[111]A. Thampi. "Path uploads your entire iPhone address book to its servers." Available: http://mclov.in/2012/02/08/path-uploads-your-entire-address-book-to-their-servers.html

[112]S. Thurm and Y. I. Kane, "Your Apps are Watching You," *WSJ*, 2011.

[113]E. Toch, J. Cranshaw, P. H. Drielsma, J. Y. Tsai, P. G. Kelley, J. Springfield, L. Cranor, J. Hong, and N. Sadeh, "Empirical models of privacy in location sharing," In Proc. *UbiComp*, 2010.

[114]W. Verduzco. "App Ops Brings Granular Permissions Control to Android 4.3." Available: http://www.xda-developers.com/android/app-ops-brings-granular-permissions-control-to-android-4-3/

[115]T. Vidas, N. Christin, and L. Cranor, "Curbing android permission creep," *Proceedings of the Web,* vol. 2, 2011.

[116]R. Want, A. Hopper, and J. Gibbons, "The active badge location system," *ACM Trans. Inf. Syst.,* vol. 10, pp. 91-102, 1992.

[117]Wikipedia. "App Store (iOS)." Available: http://en.wikipedia.org/wiki/App_Store_(iOS)

[118]Wikipedia, "Google Play," 2013.

[119]S. Wilson, J. Cranshaw, N. Sadeh, A. Acquisti, L. F. Cranor, J. Springfield, S. Y. Jeong, and A. Balasubramanian, "Privacy Manipulation and Acclimation in a Location Sharing Application," In Proc. *Ubicomp*, 2013.

[120]R. Wong, J. Lin, S. Amini, J. I. Hong, J. Lindqvist, and J. Zhang, "Connecting App Behaviors to User concerns: Improving User Interfaces for Mobile App Privacy," In Proc. *under review*, 2013.

[121]T. Yan, V. Kumar, and D. Ganesan, "CrowdSearch: exploiting crowds for accurate real-time image search on mobile phones," In Proc. *MobiSys*, 2010.

[122]L. Yang, N. Boushehrinejadmoradi, P. Roy, V. Ganapathy, and L. Iftode, "Short paper: enhancing users' comprehension of android permissions," In Proc. *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, 2012.

[123]Y. Zhou, X. Zhang, X. Jiang, and V. W. Freech, "Taming Information-Stealing Smartphone Applications (on Android)," In Proc. *TRUST*, 2011.

[124]C. Ziegler. "Google unveils 'Bouncer' service to automatically detect Android Market malware." Available: http://www.theverge.com/2012/2/2/2766674/google-unveils-bouncer-service-to-automatically-detect-android-market

APPENDIX. A

Top 11 most sensitive and frequently used permissions mentioned in Chapter 4.1

Permission	Permission Group	Description
INTERNET	Network communication	full Internet access
READ_PHONE_STATUS		
ACCESS_FINE_LOCATION	Your location	fine (GPS) location
ACCESS_COARSE_LOCATION	Your location	coarse (network-based) location
READ_CONTACT	Your personal information	read contact data
GET_ACCOUNTS	Your accounts	discover known accounts
READ_SMS	Your messages	read SMS or MMS
SEND_SMS	Your messages	send SMS or MMS
BLUE_TOOTH	Network communication	create Bluetooth connections
CAMERA	Hardware controls	take pictures and videos
RECORD_AUDIO	Hardware controls	record audio

APPENDIX. B

The list of apps and relevant permissions that covered in Chapter 5.3

App Name	Permission	App Name	Permission
Antivirus	ACCESS_FINE_LOCATION	Air Control Lite	ACCESS_COARSE_LOCATION
Brightest Flashlight Free	ACCESS_FINE_LOCATION	Angry Birds	ACCESS_COARSE_LOCATION
Compass	ACCESS_FINE_LOCATION	Antivirus	ACCESS_COARSE_LOCATION
Coupons	ACCESS_FINE_LOCATION	Brightest Flashlight Free	ACCESS_COARSE_LOCATION
Dolphin Browser HD	ACCESS_FINE_LOCATION	ChompSMS	ACCESS_COARSE_LOCATION
Earth	ACCESS_FINE_LOCATION	Compass	ACCESS_COARSE_LOCATION
Evernote	ACCESS_FINE_LOCATION	Coupons	ACCESS_COARSE_LOCATION
Facebook	ACCESS_FINE_LOCATION	Dictionary	ACCESS_COARSE_LOCATION
Foursquare	ACCESS_FINE_LOCATION	Dolphin Browser HD	ACCESS_COARSE_LOCATION
GasBuddy	ACCESS_FINE_LOCATION	Earth	ACCESS_COARSE_LOCATION
Goggles	ACCESS_FINE_LOCATION	eBuddy	ACCESS_COARSE_LOCATION
Google Sky Map	ACCESS_FINE_LOCATION	Evernote	ACCESS_COARSE_LOCATION
Lookout - antivirus	ACCESS_FINE_LOCATION	Foursquare	ACCESS_COARSE_LOCATION
Maps	ACCESS_FINE_LOCATION	GasBuddy	ACCESS_COARSE_LOCATION
Movies	ACCESS_FINE_LOCATION	Goggles	ACCESS_COARSE_LOCATION
myYearbook	ACCESS_FINE_LOCATION	Horoscope	ACCESS_COARSE_LOCATION
Seesmic	ACCESS_FINE_LOCATION	Lookout - antivirus	ACCESS_COARSE_LOCATION
Shazam	ACCESS_FINE_LOCATION	Maps	ACCESS_COARSE_LOCATION
Skyfire Web Browser	ACCESS_FINE_LOCATION	myYearbook	ACCESS_COARSE_LOCATION
The Weather Channel	ACCESS_FINE_LOCATION	Seesmic	ACCESS_COARSE_LOCATION
Toss It	ACCESS_FINE_LOCATION	Shazam	ACCESS_COARSE_LOCATION
Twitter	ACCESS_FINE_LOCATION	Skyfire Web Browser	ACCESS_COARSE_LOCATION
WeatherBug	ACCESS_FINE_LOCATION	Skype	ACCESS_COARSE_LOCATION
WhatsApp	ACCESS_FINE_LOCATION	google Street View	ACCESS_COARSE_LOCATION
Antivirus	READ_CONTACTS	The Weather Channel	ACCESS_COARSE_LOCATION
Backgrounds HD Wallpapers	READ_CONTACTS	TiKL-touch to talk	ACCESS_COARSE_LOCATION
Barcode Scanner	READ_CONTACTS	Toss It	ACCESS_COARSE_LOCATION
ChompSMS	READ_CONTACTS	TuneIn Radio	ACCESS_COARSE_LOCATION
Evernote	READ_CONTACTS	TweetCaster	ACCESS_COARSE_LOCATION
Facebook	READ_CONTACTS	WeatherBug	ACCESS_COARSE_LOCATION
Foursquare	READ_CONTACTS	WhatsApp	ACCESS_COARSE_LOCATION
GO Launcher EX	READ_CONTACTS	KakaoTalk Messenger	READ_PHONE_STATES
GO SMS Pro	READ_CONTACTS	Live Holdem Pro	READ_PHONE_STATES
Google Voice	READ_CONTACTS	Lookout - antivirus	READ_PHONE_STATES
Handcent SMS	READ_CONTACTS	Yahoo Mail	READ_PHONE_STATES
KakaoTalk Messenger	READ_CONTACTS	Google Maps	READ_PHONE_STATES
LauncherPro	READ_CONTACTS	Mouse Trap	READ_PHONE_STATES
Lookout - antivirus	READ_CONTACTS	Movies	READ_PHONE_STATES
Google Maps	READ_CONTACTS	Myspace	READ_PHONE_STATES
Pandora	READ_CONTACTS	myYearbook	READ_PHONE_STATES
Ringdroid	READ_CONTACTS	Pandora	READ_PHONE_STATES
Skype	READ_CONTACTS	Paradise Island	READ_PHONE_STATES
Tango voice & video calls	READ_CONTACTS	Real BlackJack	READ_PHONE_STATES
TiKL-touch to talk	READ_CONTACTS	Restaurant Story	READ_PHONE_STATES
Twitter	READ_CONTACTS	ROM Manager	READ_PHONE_STATES
WhatsApp	READ_CONTACTS	Seesmic - Manage Your	READ_PHONE_STATES
Wordfeud FREE	READ_CONTACTS	Social Networks	READ_PHONE_STATES
Words with friends Free	READ_CONTACTS	Shazam	READ_PHONE_STATES
Zedge Ringtones & Wallpapers	READ_CONTACTS	Skyfire Web Browser	READ_PHONE_STATES
Zynga Poker	READ_CONTACTS	Skype	READ_PHONE_STATES
Alchemy	READ_PHONE_STATES	Slice It!	READ_PHONE_STATES
Angry Birds	READ_PHONE_STATES	Talking Tom Cat Free	READ_PHONE_STATES
Ant Smasher	READ_PHONE_STATES	Tango voice & video calls	READ_PHONE_STATES
Antivirus	READ_PHONE_STATES	TapFish	READ_PHONE_STATES
Backgrounds HD Wallpapers	READ_PHONE_STATES	TiKL-touch to talk	READ_PHONE_STATES
Bakery Story	READ_PHONE_STATES	Toss It	READ_PHONE_STATES
Bible	READ_PHONE_STATES	TuneIn Radio	READ_PHONE_STATES
Blast Monkeys	READ_PHONE_STATES	TweetCaster	READ_PHONE_STATES
Brightest Flashlight Free	READ_PHONE_STATES	WeatherBug	READ_PHONE_STATES
Bubble Blast 2	READ_PHONE_STATES	WhatsApp	READ_PHONE_STATES
ChompSMS	READ_PHONE_STATES	Wordfeud FREE	READ_PHONE_STATES
Coupons	READ_PHONE_STATES	Words with friends Free	READ_PHONE_STATES
Dictionary	READ_PHONE_STATES	World Newspapers	READ_PHONE_STATES

Drag Racing	READ_PHONE_STATES	Zoo Club	READ_PHONE_STATES
Evernote	READ_PHONE_STATES	Zynga Poker	READ_PHONE_STATES
Facebook	READ_PHONE_STATES	Handcent SMS	READ_PHONE_STATES
Tiny Flashlight + LED	READ_PHONE_STATES	Horoscope	READ_PHONE_STATES
FxCamera	READ_PHONE_STATES	Inotia3: Children of Carnia	READ_PHONE_STATES
GasBuddy	READ_PHONE_STATES	Jewels	READ_PHONE_STATES
GO SMS Pro	READ_PHONE_STATES	Google Voice	READ_PHONE_STATES