A Large Scale Study on the Interplay between Users Behaviors, Expectations and Attitudes with Android Permissions

by

Weicheng Cao

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

A Large Scale Study on the Interplay between Users Behaviors, Expectations and Attitudes with Android Permissions

Weicheng Cao
Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto
2020

We recruited 1,780 participants using mobile advertising across 10 countries to study user behaviors, expectations and attitudes towards Android permissions. Participants were directed to install an Android application we developed that collected data via in-situ surveys and behavioral monitoring using Android APIs over a 30 day period. We observe how often participants grant and deny permission requests and discover some factors that driver their decisions. We also study which permissions a smartphone user expects applications to request, compute the accuracy of these expectations and their effect on our participant's permission granting and denying behavior. Then we measure participants' attitudes towards privacy and study the effect of this on their permission behavior. Lastly, we explore the effect of Covid-19 on participants' behaviors and attitudes by comparing data collected from participants that finished before and after Covid-19 breakout.

Acknowledgements

I want to thank my supervisor, Professor David Lie, for being always there to provide guidance and support for this project, and my entire term as a master student.

I would also like to thank my colleagues in this project, Steven Xia, Nina Taft, Sai Teja Peddinti and Professor Lisa Austin, for bringing in their vast experience and novel ideas. This project would not be in its current shape without their supports and contributions.

I want to also thank University of Toronto and the Department of Electrical and Computer Engineering for giving me this opportunity to conduct research and learn. I spent both my undergraduate and graduate lives here and I will cherish those memories.

Last but not least, I want to thank my family for their continuous support on and interests in my research, my study and my life.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Permission requests in the Android system provides two important functions. First, they allow users to restrict mobile application's access to resources and data on their phones. Second, they are a mechanism that informs users about the types of data that a mobile application might access. An important ramification of this system is that app developers could interpret users' decisions and find motivations towards developing privacy friendly applications. While many factors can influence users' decisions about which permissions they grant and why they deny, this behavior could nevertheless be viewed as an opportunity to learn about unpopular permissions, confusing permissions, which ones they are comfortable granting, and which explanations affect users' decisions. In our study, we focus on the "Dangerous" permissions, which must be explicitly granted by the users to the application. These permissions are categorized into 11 permission groups (such as *Location*, *Camera*, *Microphone*, etc.). For simplicity, we will refer to these permission groups as permissions. The release of each new Android update in the last 3 years has been accompanied by changes to the permission system. Prior to Android 8.0, if an app requested a permission at runtime and the user granted that permission, the Android permission system would also incorrectly grant the app the rest of the permissions that belonged to the same permission group if they were registered in the manifest. In Android 8.0, this behavior has been corrected so that the app is only granted the permissions it has explicitly requested [4]. Android 9.0 limits the ability to access the microphone or camera for apps running in the background. It also introduced the `CALL_LOG` permission group, which contains `READ_CALL_LOG`, `WRITE_CALL_LOG` and `PROCESS_OUTGOING_CALLS` permissions that belonged to the `PHONE` permission group previously. This offers users better control and more visibility of the apps that require access to sensitive phone call information [4]. Android 10 introduces the tri-state of *Location* permission by adding a `ACCESS_BACKGROUND_LOCATION`. The newly added state only affects an app's access to location information when running in the background [4]. We can see that, as Android system progresses, it also puts more measures in place to protect user's privacy against apps.

Many considerations and factors affect how users interact with Android permissions, such as behaviors, expectations, explanations offered, and attitude towards privacy. Prior work usually focuses on one aspect of users at a time, such as behaviors [16, 29, 48], expectations [47, 27, 33] or attitudes [39, 27]. However, none of these seek to analyze the interplay of these factors over the same set of users in a single study. Moreover, it is preferable to obtain behavior data "in-the-wild", when users employ their own devices as opposed to experiments in a lab, as this captures more naturally the choices users make in their daily lives. Our study is across several geographic regions with the magnitude of thousands on

a global scale.

In order to overcome these challenges, we designed an Android app, called PrivaDroid, and used this as our study instrument. It is designed to run in the background on participants' phones. It observes any app install, app removal, permission grant and permission deny event, and launches an in-situ survey immediately after the event. Together, the observations and surveys collect data on participant attitudes, behaviors and expectations at the moment they act on their own personal devices. To enable us to reach a broad base of participants, we design PrivaDroid to support all major Android versions from 6.0 to 10, translate PrivaDroid into 4 major languages and use mobile advertising to recruit participants.

Our collection of decision rationales is similar to [16]; in fact, we re-use the questions from this prior study, so rationales for permission decisions can be directly compared. We expand beyond the prior study in multiple ways: 1) the prior study was done with US based participants only, whereas our study includes participants from 10 countries and regions, and our app was deployed in 4 languages; 2) we collect information about which permissions a user expects an app to ask for and thus can compare expectations against behaviors; 3) we separate apps that provide explanations for their permission requests from those that do not, and can thus assess the impact on deny rate of providing explanations; and 4) we have users complete a privacy attitudes survey at the end of our study, so that we may compare self-stated privacy sensitivity with actual behavior.

The app was published on the Google Play Store and advertised on several online advertising platforms to recruit participants. To the best of our knowledge, this is the first cross-continent study on Android permission decision making. Each participant used our app throughout a one month period. We selected the 10 countries in order to study the potential differences caused by cultures, languages and geographic locations in user behaviors around and privacy attitudes towards individual permissions. We first rolled out PrivaDroid in the five English speaking countries (i.e. US, Canada, UK, India and South Africa). Before we released PrivaDroid for the other five countries, Covid-19 broke out. Despite its catastrophic effects on the world, this presented to us as an opportunity to explore how an epidemic like this influences user behavior around apps asking for permissions and user's privacy attitude. Therefore, we started a second round of advertising for the five English speaking countries in order to compare the data collected from the same set of countries before and after Covid-19 while recruiting participants from the other five countries to complete our main experiment. Over the course of our main experiment, 1,780 participants (38% females) joined from 10 countries and successfully finished the 30 day study. In total, we observed 74,381 app installs, 67,094 app removals, and 36,095 permission decision events. Nearly 1/3rd of these events have been surveyed. This is a much larger scale study than [16] which was based on 157 participants. As for our experiment to study Covid-19 impact on user behavior, in total we have 1599 participants (42% females) who finished the study before (*Pre-Covid* group) and after Covid-19 (*Post-Covid* group). We saw 56,689 app installs, 50,595 app removals and 39,229 permission decision events. The scale of the data is unprecedented.

Prior studies have advocated that explaining the reasons for permission requests to users is critical to improve their understanding, which in turn influences their grant and deny choices [35, 27, 31]. In previous surveys, users state they would be more comfortable granting permissions if explanations were offered [40]. As a result, users may seek explanations from apps for their permission requests before granting. In our study, we capture any rationale messages explaining why an app requires a permission and evaluate how this affects users' permission grant/deny decisions.

1.1 Contributions

Our contributions can be summarized as follows.

- We designed an Android app to do experience sampling, that we translated into Spanish, French and Chinese (Traditional). We demonstrated that it is possible to use online advertising instead of in-person meetings or Amazon Mechanical Turk to recruit participants around the world.
- We compare the deny rate trends today to the study done three years ago [16] and report which trends have stayed the same (e.g., aggregate deny rates), and which have evolved (e.g., the behavior differences across genders). For example, deny rates have increased for some permissions such as *Microphone* and *Calendar*, however they have stayed the same for the most frequently requested ones, such as *Location* and *Storage*.
- Our analysis across countries and regions revealed that the intra-country variance of permission deny rates is much higher than the inter-country variance. This suggests that differences across countries may be less pronounced than often assumed.
- We study user expectations of permission requests and find that in general users predict which permissions an app will request incorrectly 69-83% of the time depending on the permission. Moreover, we do see expectations influencing behavior, since when faced with an unexpected permission request, our participants deny rate increased by 16%. We confirm that apps that provide explanations to users for permission requests experience lower deny rates than those that don't, illustrating that explanations are beneficial to both user and app developers.
- We compare attitudes to behaviors and see that ∼25% (288/1171) of our participants who say they are privacy sensitive do in fact have low deny rates. We further analyze this set of participants who exhibit this behavior and and compare them with those who don't, thus shedding new light on the well known "privacy-paradox" in the context of Android permissions.
- We evaluate the influences of Covid19 on user behavior of participants from the five English speaking countries (i.e. US, Canada, UK, India and South Africa). We discover that the aggregate deny rate of participants who joined after Covid-19 breakout than those before.

1.2 Thesis Structure

The rest of the paper is organized as follows. Section 2 explains the concepts of Android and technologies used in our experiment. Section 3 discusses related work. Section 4 explains the participant recruitment method, while Section 5 describes the design, data collection and implementation of our PrivaDroid app. Our findings are presented in Section 6. We discuss our findings in the analysis of data collected from the participants from the *Pre-Covid* and *Post-Covid* groups. Section 8 describes the limitations of our study and Section 9 discusses the future works. Section 10 concludes the paper. All the survey questions are listed in Appendix A.2.

Chapter 2

Background

We included the required knowledge and methods we used in this chapter to help with better understanding of this thesis. Our app, PrivaDroid, is an Android application written in Java. Its purpose is to observe app install, app removal and permission decision events. In order to complete those goals, PrivaDroid taps into various Android APIs and services. This chapter also talks about mobile advertising and its applications in the current day. We used multiple mobile advertising platforms to recruit participants for our mobile study.

2.1 Android

2.1.1 Permissions Groups

Permissions in Android are categorized into different groups. Each permission group is related to some device's features or capabilities [4]. For example, the SMS permission group has both `READ_SMS` and `RECEIVE_SMS` permission declarations. `READ_SMS` allows applications to read text messages that users have sent and received whereas `RECEIVE_SMS` enables applications to directly receive messages. Although Android has such fine-grained permission classes, permission requests are handled at a permission group level. When an app requests a specific permission, such as `READ_SMS`, Android system will ask for user's permission without explicitly mentioning the app is requesting `READ_SMS` permission but only access to SMS group. Once the user grants a permission within a permission group, the app will have access to other permissions in the same group as well.

2.1.2 Permission Protection Levels

Permissions fall into different protection levels, namely *normal*, *signature*, and *dangerous* permission. *Normal* permissions are those that allow the app access to isolated application-level features. They pose little risk to user's privacy or operations of other apps. An example of a *normal* permission is for an app to set the time zone. Android system automatically grants the *normal* permissions at installation. Another type of protection level is *signature* permissions. A *signature* permission is defined to be a permission that Android system grants only if the application requesting that permission is signed by the same certificate as the application that declared that permission. The third type, *dangerous* permissions, is our focus in our study. *Dangerous* permissions consist of 11 permission groups including *Location*,

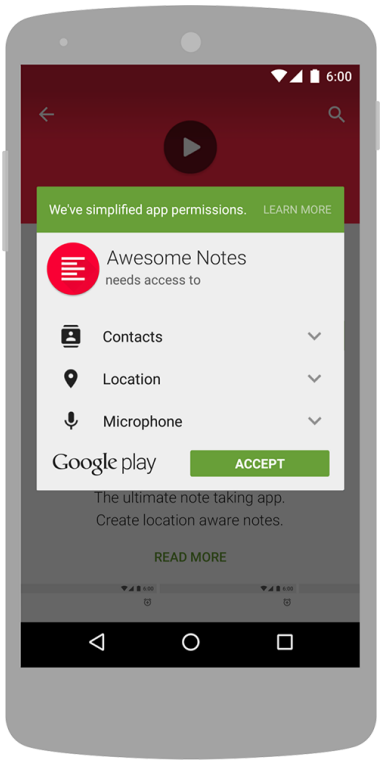

Figure 2.1: Install-time permission dialog [4]

Microphone, and etc. They often involve access to users' private information, or could potentially impact the user or operations of other apps. For simplicity, we will refer them as just permissions in this paper.

2.1.3 Requesting Permissions

Only the aforementioned *dangerous* permissions are required to be explicitly granted by the users. For apps running on devices with versions lower than Android 6.0 or apps that target versions lower than Android 6.0, users must grant all *dangerous* permissions at installation of the apps. Figure 2.1 shows the install-time permission dialog, in which if the user accepts, Android system will grant all permissions the app requests.

Starting in Android 6.0, users are not required to grant all permissions when installing an app. Alternatively, users are asked to grant the dangerous permissions at runtime. When an app tries to access a *dangerous* permission that has not been granted before, Android system will prompt a permission dialog like the one in Figure 2.2. The dialog contains information such as the app name, which permission group the app is requesting and users can grant or deny this permission request. In order to prevent permission requests from being denied repetitively, on the dialog there will be an option for users to suppress further requests. This option is controlled by Android system, not the app itself.

Android 6.0 and above also allow users to change their permission settings in the Settings menu. Users can go into the individual app's information screen and toggle the permissions of that app. For Android 8.0 and above, apps that request each permission are grouped together to provide users with a more convenient way of bulk managing a permission. Figure 2.3 shows the two locations where users can change permissions.

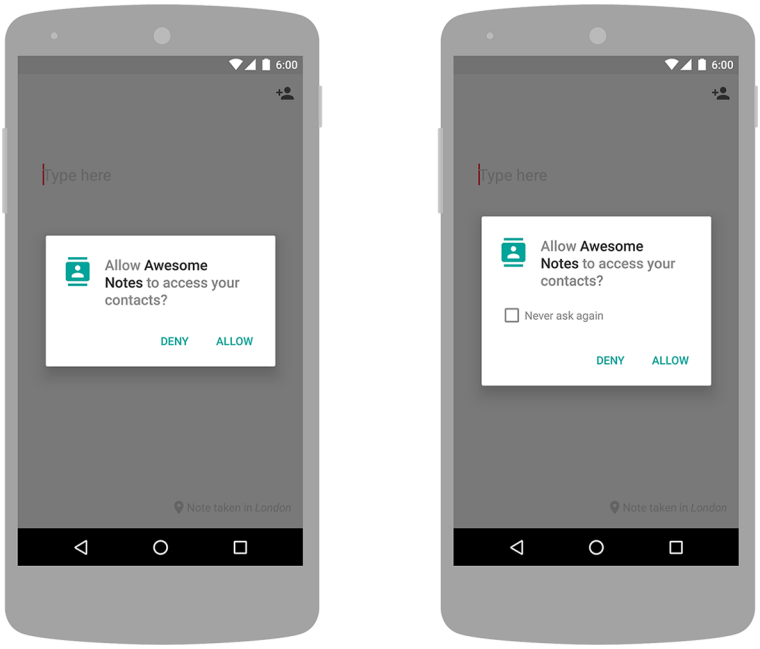

Figure 2.2: The first permission dialog (left) and subsequent permission request with option to turn off further requests (right) [4]

In this paper, we define a *runtime permission request* to be the system permission request shown in 2.2. *Permission grant* events include both runtime permission requests that users granted (i.e. users pressed "ALLOW" on the permission request dialogs) and permissions users granted in the Android Settings menu. Similarly, *permission deny* events consist of runtime permission requests users denied and permissions users turned off in the Settings menu.

2.1.4 Accessibility Service

Accessibility in Android is to assist users with disabilities. It consists a wide range of functionalities, such as screen reader and interaction control. Accessibility services run in the background in Android system and receive callbacks when an `AccessibilityEvent` is fired. Events like a change of focus and a button click denote a state change or transition and will trigger an `AccessibilityEvent`. There is often a tree of `AccessibilityNodeInfo`s representing a snapshot of a View state associated with an `AccessibilityEvent` [4].

Android also provides `AccessibilityService` APIs to encourage developers to create more accessible apps by allowing apps to register for these accessibility events. Accessibility service is used in PrivaDroid in order to detect runtime permission dialogs and participants' decisions on those dialogs. Its implementation details are discussed in Section 5.

2.1.5 App Usage

Android provides an API to query the usage data for an individual app package. This usage data can be accessed by `getSystemService()` with `USAGE_STATS_SERVICE` parameter to obtain the `UsageStatsManager` [4]. After detecting a runtime permission request, we query the usage manager to extract the last active app so that we know which app initiated the request. Apart from recording participant's decision on

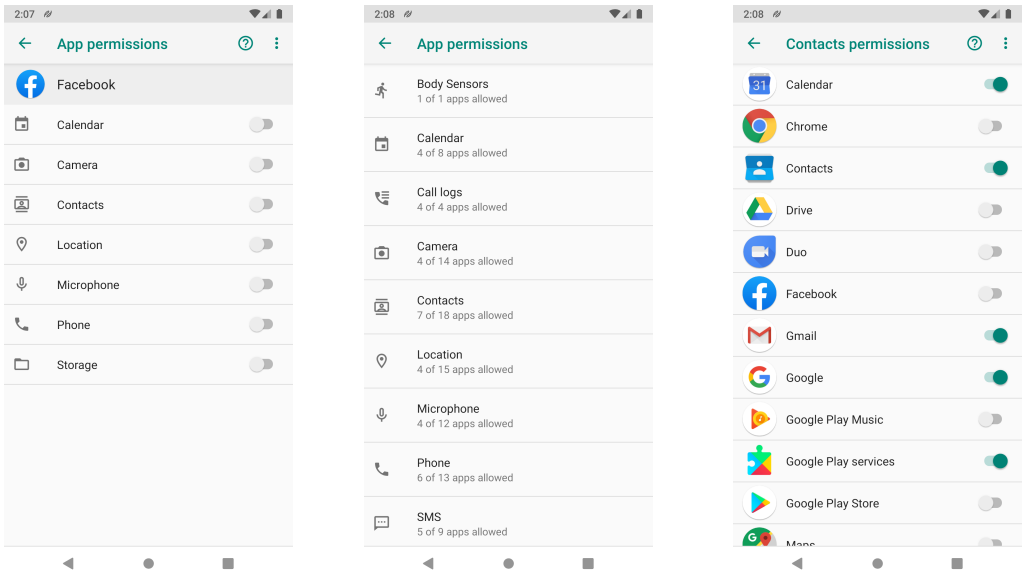

Figure 2.3: Permission settings of an individual app (left) and apps that request each permission grouped together (right two)

the permission request, we also log the total time the participant spent on this app during the last hour before the request as well during the last 30 days.

2.1.6 Android Services

A *Service* is a component Android applications can use to perform a long-running action that does not require interaction with the user. There are three different types of services in Android, namely *foreground service*, *background service*, and *bound service*.

A *foreground service* is usually used when applications perform some actions that are noticeable to the user. For example, a music player app can use *foreground service* to play a music track. A *foreground service* must display a notification to notify the user that it is running. It can continue to be running even when the user is not interacting with the app. We implemented a *foreground service* that registers a broadcast receiver to monitor app install and app removal events. This foreground service is always running on participants' devices.

A *background service* is a service that runs in the background and is not directly noticed by the user. For example, an app could perform an asynchronous query to a cloud database using a background service and it is not necessary to make this operation visible to the user. We could not use a *background service* to register the broadcast receiver that receives system intents for app install and app removal events. This is because of the system restrictions introduced in Android 8.0 (API level 26) on running background services when the app is not in the foreground. From Android 8.0, Android does not allow a background app to create a background service anymore.

The other type of service is *bound service*, which allows different Android components to interact with it by providing a client-server interface. The available interactions include sending requests, receiving result and interprocess communication (IPC). We only used *foreground service* in our implementation of PrivaDroid.

2.2 Google Advertising ID

Google advertising ID is a unique device ID for advertising purpose, provided by Google Play services [3]. It allows Android users to better control their online privacy and provides app developers with a central system to continue monetizing their apps. Although Google advertising ID is used primarily for advertising purposes, we used it as a unique participant identifier by tagging the events sent from the same device with this ID.

2.3 Mobile Advertising

Mobile advertising is an advertising method involving advertising material built specifically to appear on a mobile phone. There are mainly two types of mobile advertising [10], in-app advertising and web-based advertising.

In-app advertising means app developers serve advertisements in their apps. Video ads and display ads are two major formats of in-app advertising. For example, gaming apps can integrate video ads and reward users with virtual goods or currency in return for watching a video ad [8]. Another demonstration of video ads is social apps, such as Instagram and Facebook. Both apps allow merchants to create a video to promote their products or display a static image for an amount of time in the app. As for display ad formats, there are native ads, which fit the native look of the app and placed natively in the app, and banner ads, which are ubiquitous in free apps.

Web-based mobile advertising is similar to regular web advertising on desktop. Websites are optimized for mobile because of the reduced screen size and so are the advertisements on those sites. Both video-based ads and display ads are used in web-based mobile advertising. An example of display ads on mobile websites is Google search. When mobile users search via Google, there can be an ad banner appearing on top of the search results. We chose Google, Facebook and Reddit as our advertising providers and discussed the specific configurations such as targeted audience and ad formats in Section 4.1.

Chapter 3

Related Work

This is an extensive amount of existing research in the space of Android permissions and user privacy. While it would be almost impossible to cover the entire body of work, this chapter aims to provide an overview of the users' privacy expectation with permissions, apps providing explanation for requesting permission, cross country studies, and privacy paradox and concentrate on the works most related to our experiment. We begin by looking at users' understanding of Android permissions since they are the most affected stakeholders when it comes to Android permission and privacy. While permission understanding influences user behavior around permission, research showed that whether users expected a permission request also has an effect on their decisions on permission requests. Then we move on to ways to shape users' expectation better by providing explanations of why permissions are required. Lastly, we discuss a phenomenon called "privacy paradox" that we observed in our data as well as in previous studies.

3.1 Understanding of Permissions

Research focusing on Android users has shown that few users actually read application permission requests and fewer understand them [20, 24]. The authors of [20] conducted two install-time permission system usability studies, an Internet survey of 308 Android users, and a in-lab study of 25 Android users in. In the Internet survey, participants were asked what they looked at when deciding to install an app. Only 18% of the respondents reported paying attention to the permissions. Participants also answered three randomly-selected quiz questions from a set of multiple-choice questions regarding the abilities each Android permission allows. For example, participants were required to select what `READ_PHONE_STATE` permission allows apps to do. Available options are "Read your phone number", "See who you have called", "Tract you across applications", "Load advertisements", "None of these", and "I don't know". The result shows that only 2.6% of the respondents answered all three questions correctly. This indicates that few users understand the permissions let alone benefit from the Android permission system [20]. We observed the same phenomenon in our study. Instead of using self-reported survey, we directly measure users' understanding of permissions by observing their behaviors. When installing an app, we asked the participants to predict what permissions the app may require. We then compared what participants predicted with what permissions were actually asked for by the app to calculate the correct guess rate. We discovered that the overall average probability of users correctly guessing the permissions an app requires is only 30%.

3.2 Privacy Expectation with Permissions

Research on user privacy expectations with permissions has shown that users are concerned when they learned of the possible risks associated with permissions [19], and are surprised by the applications' ability to collect data when running in the background [23, 42]. In [27], the authors studied users expectations around 4 resources (GPS location, Device ID, network location, contact list) based on an older model of Android. Specifically, they designed a Human Intelligence Task (HIT) on Amazon's Mechanical Turk (AMT) to ask a few questions about a specific Android app and resource pair. Each HIT consists of a few screenshots and app description, both retrieved from the official Google Play Store. MTurk participants were shown one of the two sets of questions. The first set, referred to as *the expectation condition*, was designed to capture participants' expectation of a given app accessing a sensitive resource and reasons they think the app required that resource. The other set, referred to as *the purpose condition*, measures the comfort level when given specific reasons why the resource would be accessed. They discovered that users are more uncomfortable about apps accessing sensitive resources when encountering unexpected app behaviors. This study captured resource requests users did not expect via an mTurk survey, not based on decisions on personal devices as in our study. In our study, we measured participants' expectation after they made a decision on a runtime permission request. We discovered that participants were more than twice likely to deny a permission request when they did not expect it. In addition, we also measured users' comfort level when users granted a permission and confirmed that users are more uncomfortable when granting an unexpected request.

Wijesekera et al. [47] captured user expectations by monitoring their apps for one week and showing users afterwards what was collected and asking in-lab questions about whether the participants expected that. This study reports that users said they were more likely to deny permissions they did not expect. Our results corroborate this finding, however we use a very different mechanism. We recruited participants through online advertising and let them install an app we created. This app monitors permission decisions made on runtime permission request as well as in the Android Settings menu on participants' personal devices. Our app asks participants if they expected an app to ask for a permission immediately after they made a decision on a runtime permission prompt. This approach captures participants' expectations while they still remember the context in which the permission was requested and offers a more accurate measurement of their expectations. Our experiment was also at a much larger scale.

3.3 Explanations for Permission Requests

Explaining why a permission is required can help better shape users' expectation of permission requests. To help provide explanations or additional information so users can make better choices, Harbach et.al. [21] and Kelley et.al [25] have suggested providing more privacy information and personal examples to help improve user comprehension. In [25], the authors conducted two studies: a laboratory exercise of 20 participants and a mTurk study of 366 participants. Participants were asked to role-play selecting Android applications for a friend and were assigned to refer to either a privacy checklist the authors came up with or the current permissions display in the Android market. Figure 3.1 shows the standard Android market permission display and privacy checklist display. Results show that participants who chose the privacy checklist display were more likely, on average, to select the application that requested fewer permissions. This indicates that when fully understanding the potential privacy compromise (by

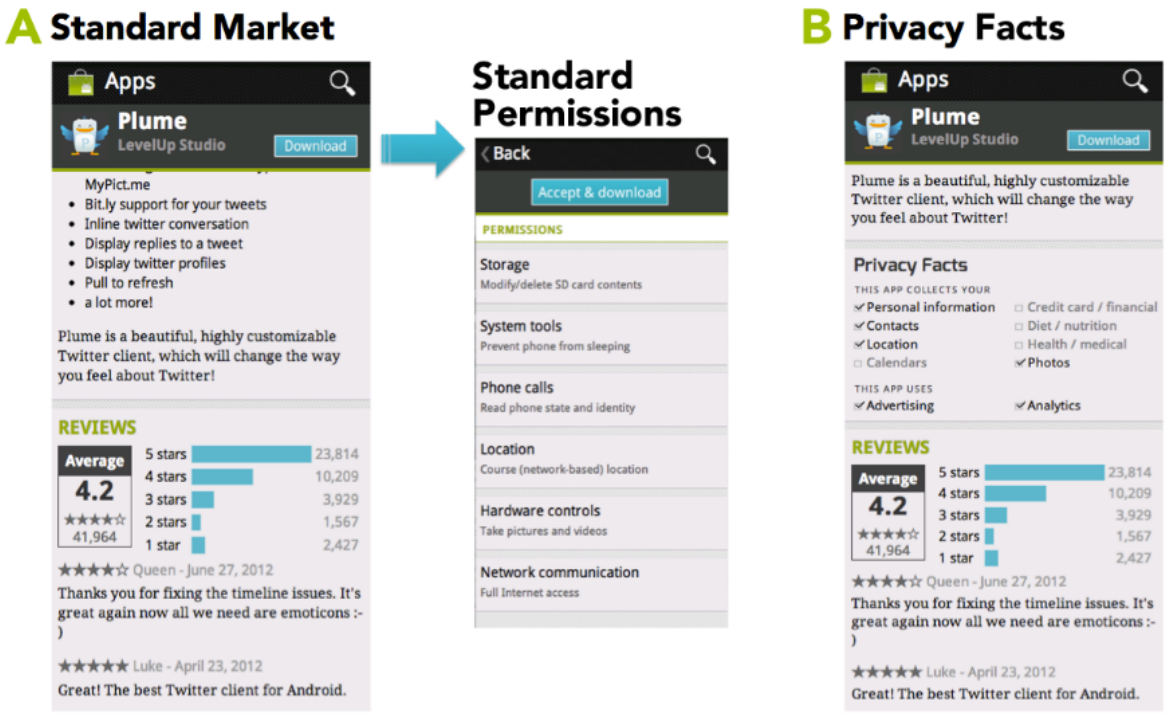

Figure 3.1: The two privacy/permissions display conditions tested

granting more permissions to apps), users are more cautious with privacy. This warrants the necessity of more explanation by developers of why permissions are required.

Others categorized permissions to reduce the number of privacy/security decisions users need to make [18]. Some have even explored creating personalized privacy assistants [29], or surfacing nudges to assist users with decision making [15]. In [15], the authors focus on developing supplementary features to help users make decisions while we focus on developer provided rationales. We monitored permission-related strings on participants' screens right before a permission request. Through a keyword-comparison heuristics method, we recorded the strings that are potentially an explanation of why this permission is requested.

Specific to the developer provided permission rationales, there is very little prior work. Tan et al. [40] conducted an online survey of smartphone users and showed that permission requests that include explanations are significantly more likely to be granted. They also analyzed ∼4K iOS apps and showed that only 19% of the permission requests included text within the dialogs to explain the resource usage. Liu et al. [30] analyzed ∼83K Android apps and the extracted permission rationale messages, and showed that less than 25% of apps provide rationales and that the purposes stated in a significant proportion of these rationales are incorrect. We have made similar observations in our analysis too: only 15% of apps in our data displayed a rationale messages to explain their permission requests, and having a rationale messages reduced the permission denial rate to 7.1% (compared to 17.3% with no message). While the prior work mentioned influence of permission rationales on the denial rates based on surveys, ours is the first study to study user behavior and quantify the reduction in the permission denial rate when there are rationales.

3.4 Cross Country Studies

There has been little research related to exploring differences in privacy attitudes on Android permissions across users in different countries. Shklovski1 et al. [39] conducted a qualitative study and then a survey across two countries (Iceland and Denmark) to investigate how smartphone users feel about data access on their phones and if they may be willing to change their behavior once they have been informed about tracking and data leakage. [34] did a multi-country survey and shows that psychographics and various attributes of the mobile app context are predictive of users' privacy preferences. On the other hand, Schubauer et al. [38] examined app behavior on the Google Play Store across three categories and 3 countries (US, South Korea and Germany) and discovered that policy changes aligned with privacy law changes (such as General Data Protection Regulation) have impact on the application permission usage. Our study is the first to perform a large scale comparison of both behavior and attitudes of Android users across 10 countries and regions.

Online advertising has been adopted by many Android user studies including those with cross country studies. The authors of [47] put an online recruitment advertisement on Craigslist in October of 2014. [20] used AdMob's Android advertising service and displayed advertisement in applications on Android devices in the U.S. and Canada to recruit participants to complete an Internet survey. For their laboratory study, [20] posted a Craigslist ad to recruit participants in the San Francisco Bay Area. Instead of using websites, we used Google, Facebook and Reddit and advertised our app in their own mobile products such as Google search, YouTube, Instagram, Facebook and Reddit Mobile.

3.5 Privacy Paradox

People can behave in a manner inconsistent with their stated privacy preferences and values, which researchers label the "privacy paradox" [36]. An example of "privacy paradox" is that users claim to be sensitive and not willing to share their private data with apps but they pay little attention and always grant apps' requests for some permissions. Prominent interpretations of the paradox draw upon behavioral economics and focus on reasons such as: incomplete information, asymmetrical information, bounded rationality, and the impact of behavioral anomalies and biases [14]. [22] provides a mechanism to help users, who may exhibit privacy paradox behaviors, to make decisions about permissions by highlighting discrepancies between general user privacy attitudes and app riskiness. A nice survey of the privacy paradox literature [26] highlights the complexity of this phenomenon and advocates that future studies should use evidence of actual behaviour rather than self-reported behaviour. This approach aligns with our methodology. We measured participants' privacy attitudes via a set of survey questions and together with their decisions on runtime permission requests and stated reasons behind their decisions, we can evaluate if a participant exhibits privacy paradoxical behaviors. In our study, we observe privacy paradoxical behaviors but we attempted to explain them from not only user behavior and privacy attitude perspectives but also participants' expectation of the permission requests point of view.

Chapter 4

Participant Recruitment

4.1 Participation Composition

In order to draw comparison between user behaviors with Android permissions across a wide range of regions and cultures, we recruited participants from 10 countries and regions: Canada, United States, United Kingdom, India, South Africa, Singapore, Spain, Argentina, France and Hong Kong. This set of countries covers 5 continents, 4 languages and includes 2 developing economies (India and South Africa). We initially included South Korea in our targeted countries. However, the advertisement uptake in South Korea was so low that very few users signed up for our experiment, and hence we excluded it from our study. It could be because there was not as much of an active audience on the advertising platforms we chose in South Korea as there was in other countries. We also consulted with students from South Korea in our research group and colleagues living in South Korea. They suggested that there are several other popular social media platforms including Line and KakaoTalk and thus our ads on the three chosen advertising platforms might not get enough attraction. Therefore, we decided to leave South Korea out of our study. We aimed to recruit at least 100 participants from each region with a nearly balanced split between males and females (this proved to be difficult in some regions). We did not control for other variables, such as age, profession or income during the user recruitment process, mainly due to the inaccuracy in the advertisement network inferred attributes for targeting our ads and partly for privacy reasons. However, we advertised to all age groups and professions on platforms where such a control was available.

4.2 Advertising and Compensation

We decided to use online advertising for our recruitment needs, in order to have a single method to recruit across many countries. Most recruitment agencies for user studies only work in a single country. We wanted the ad to target primarily at Android application users. Therefore, we selected three popular mobile online advertising providers, namely Google, Facebook and Reddit, so as to reach a broad international audience. Each of them has both mobile application products and mobile websites versions with a diverse set of users and ability to place our ads in them. We used both video ads and display ads on these platforms.

Google Ads. For Google Ads platform, we used both texts and images. They were placed on multiple

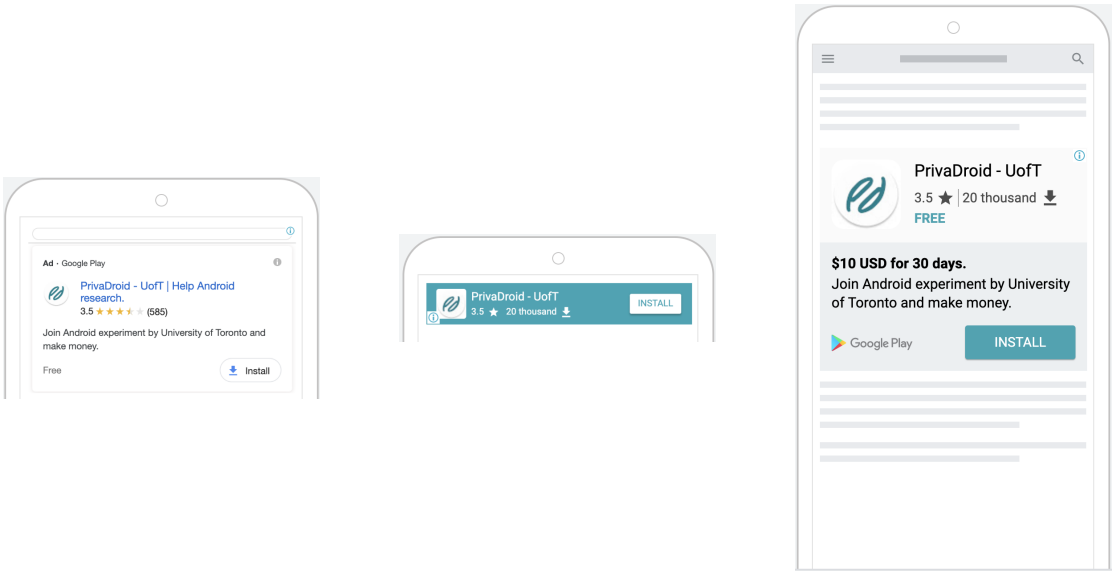

Figure 4.1: Google search result advertising placements

products including Google search, YouTube and Google Play Store. We only controlled for locations and languages of targeted audience since Google Ads does not support other demographic targeting. Figure 4.1, 4.2 and 4.3demonstrate the advertising placements on Google platforms.

Facebook Advertising. For Facebook advertising platform, we chose to place ads on their main products, namely Facebook, Messenger and Instagram. Facebook advertising supports demographic targeting such as gender and age besides country and language. We first targeted female participants from 18 to 65 and once we reached 50 female users, we started advertising for both genders with the same age group. Our advertisement material, an image containing some description of our experiment, was advertised in both video and native Facebook feed formats. Figure 4.4 shows our advertising materials placed on Facebook new feeds and Instagram story.

Reddit Advertising. Reddit advertising supports location, interests and communities targeting. Interests targeting means targeting the advertisement to the users who interacted with content with a particular interest recently. Communities targeting allows advertisers to target subscribers of specific subreddits. In order to attract more female participants first, we first practiced interests targeting by targeting a list of interests that many have more female readers. These interests include Animals & Pets, Art & Design, Style & Fashion, Food & Drinks, and Family & Relationships. We selected these interests based on a study stating that women are more interested in artistic and social interests than men are [37]. We acknowledge that this may introduce bias in female participant selection but once we reached a sufficient number of female participants, we relaxed the targeting restriction to all interests in order to dilute this bias. Figure 4.5 consists of the advertisements shown natively in the Reddit app and mobile version of Reddit website.

Among the three mobile advertising providers, Facebook offers the most diverse targeting functionalities including gender, age (although we did not target any age group specifically) and the most detailed breakdown of ad performance metrics such as install rates of each gender and age group. Initial experimentation with our app revealed that male participants were more likely to join our experiment than females. In order to improve gender balance across our participants, we decided to recruit female

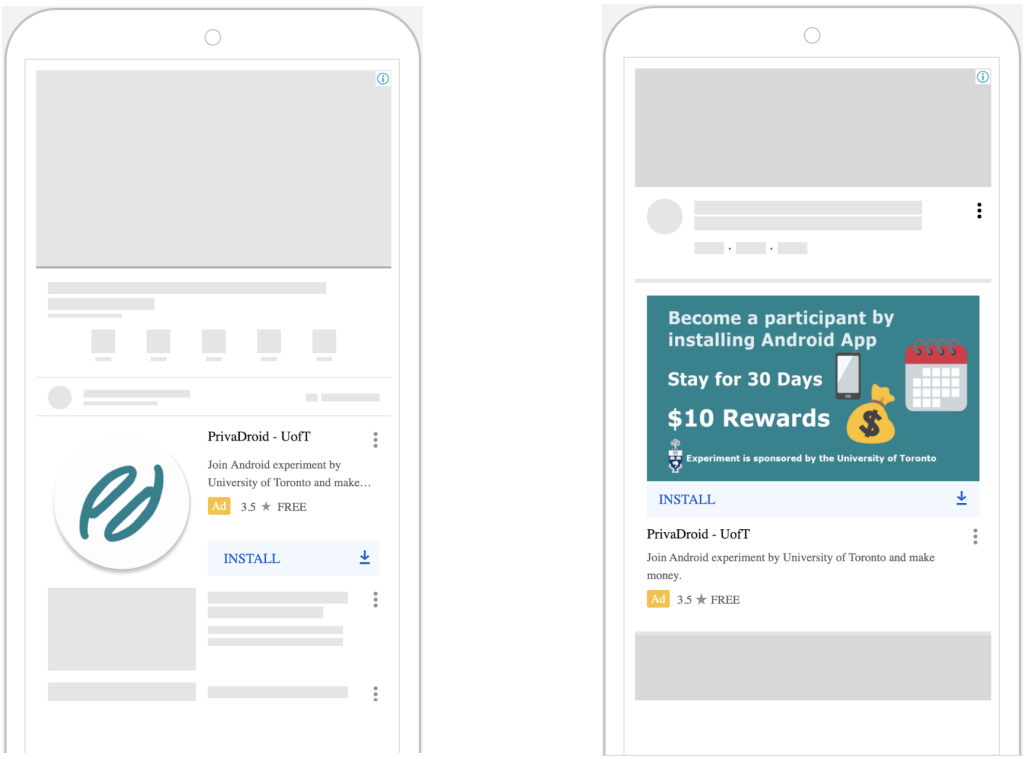

Figure 4.2: YouTube advertising placements

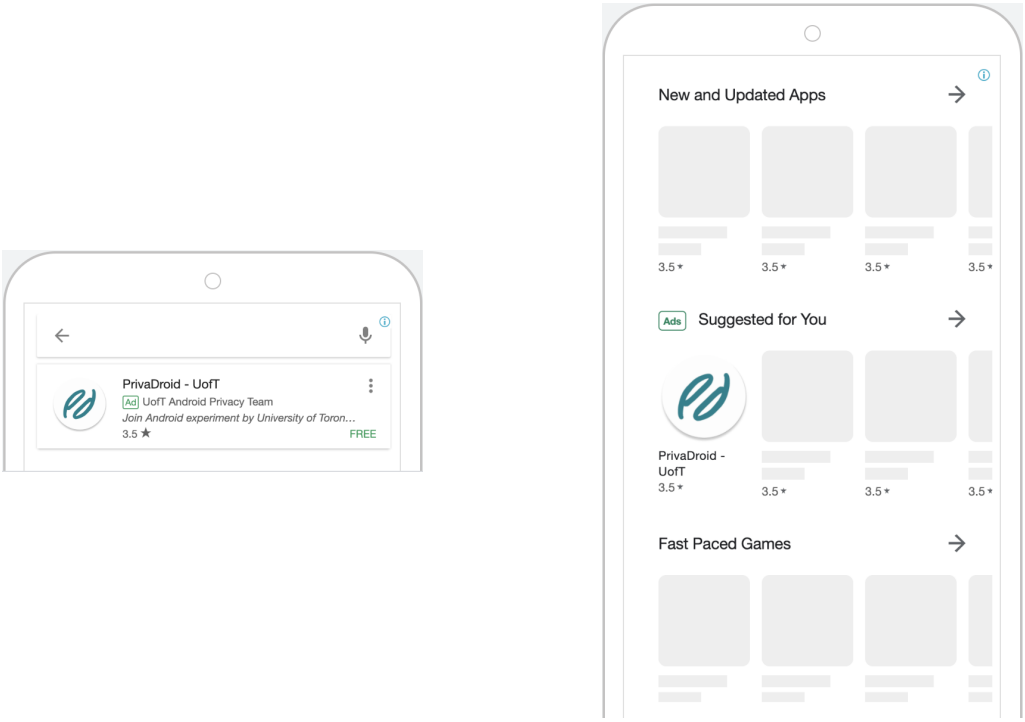

Figure 4.3: Google Play Store advertising placements

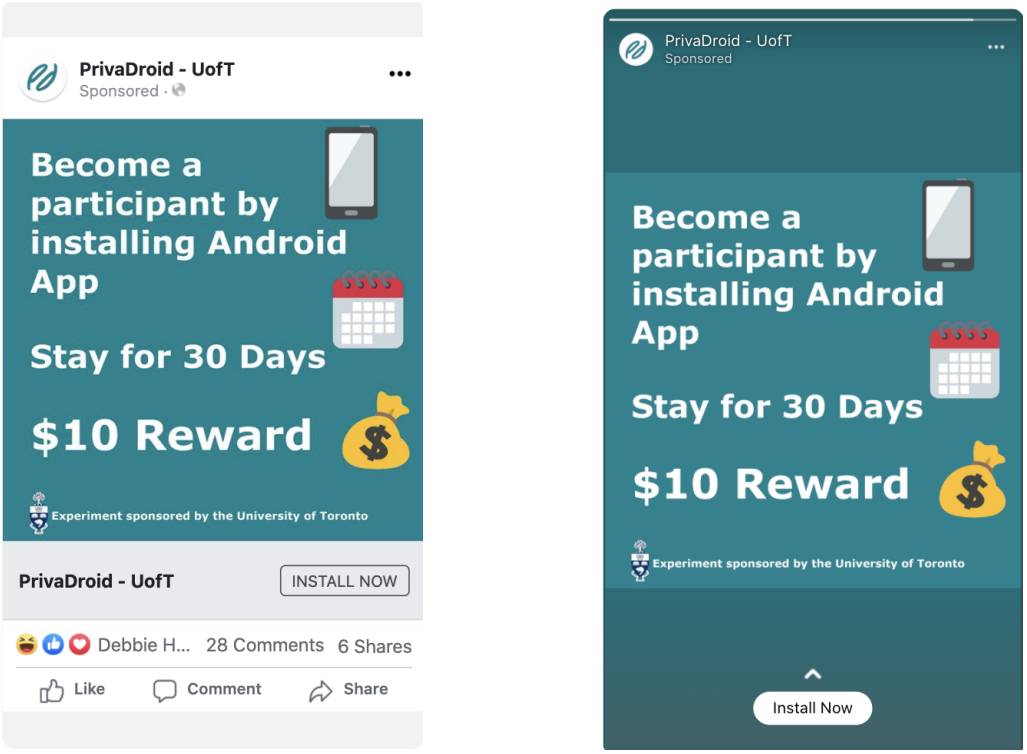

Figure 4.4: Facebook and Instagram advertising placements

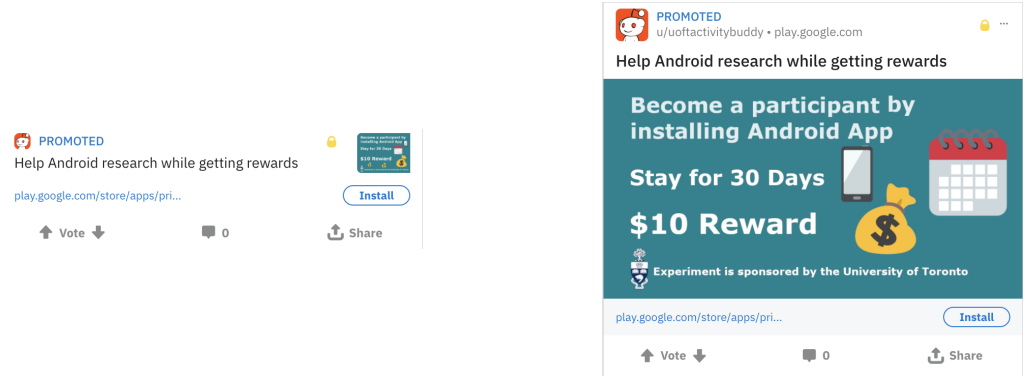

Figure 4.5: Reddit advertising placements

participants first, and only started advertising to males after we had more than 50 female participants (with the exception of Hong Kong).

The compensation to our participants is $10 USD if they complete the experiment by staying for 30 days. We initially selected Bitcoin and PayPal as our payment methods. Nevertheless, Bitcoin was not approved by our Research Ethics Board (REB), hence we used only PayPal in all 10 countries.

4.3 Transparency and User Consent

Our study was approved by the University of Toronto's REB. Participants need to give their consent before enrolling in our experiment. When participants install and open the PrivaDroid Android app, participants will be prompted the consent form, which enabled us to both gain consent and allowed us to be transparent about our practices. It contains the following key clauses. First, participants must be from one of the specified countries and must be above 18 years of age. Second, participants must keep the accessibility service and app usage access enabled for our app during the length of the experiment. Third, participants are informed that no personally identifiable information except for their Google advertising identifier (a device ID that we use to associate all the data coming from a single device) will be collected, and this advertising identifier will be used to infer any other personal information such as name, email, etc. This consent form was translated into the 4 languages and manually reviewed by native speakers to eliminate any confusion and enhance transparency. All participants consented to these clauses and we did not receive any email or other contacts from participants about questions regarding the consent form.

4.4 Data Protection

In order to protect user privacy, access to the collected raw data is restricted to only the members of this research project at the University of Toronto. All of the user data was hosted on a Google Firebase cloud storage, not on participant's phone. Participants can view, on their devices, their data including Google advertising identifier, join date, events that we recorded, survey responses, and etc., which is all dynamically queried from the Firebase datastore.

Chapter 5

App Description

The PrivaDroid data collection platform consists of an Android application, a Firebase datastore that stores data and a flask server that analyzes the data. PrivaDroid is designed to run on participant's phone in the background, collect both behavioral data on certain events and in-situ survey responses right after those events occur. It monitors the app install and removal events by receiving system broadcast upon these events and detects participants' decisions on runtime permission requests through an accessibility service. PrivaDroid manages and tracks participant participation over the course of the study and has been localized into several languages. Participants can view the collected data in the app and submit responses to the various surveys.

5.1 Origin of PrivaDroid

PrivaDroid was inspired by an Android app called Paco, which was used in [16] to conduct a similar experiment as ours back in 2017. We did not re-use Paco as we identified several inefficiencies of Paco and eventually decided to implement our own solution with improvements on three fronts. First, Paco, which came into existence 8 years ago, was designed to be a platform for user behavior experiments in general, rather than tailored for an experiment of our kind specifically. For Paco, researchers can use an experiment configuration website run by Paco developers and publish experiments to a defined set of participants or to the public. The authors in [16] needed to modify the original Paco app in order to be monitoring app install, app removal and permission request events. As a result, there was a large portion of unused and outdated code that required maintenance and refactoring due to the Android SDK deprecation. In order for Paco to run on later Android versions, we spent a large amount of time updating and removing the unnecessary code, which was very inefficient. Second, the pipeline of Paco consists of a front-end Android application, a Java servlet application and a relational database hosted on Google Cloud Platform. In order to use Paco, we need to use Paco's backend and database, which, similar to the Paco app, required maintenance as well. The majority of the code is related to database operations and data analytics. After considerate evaluation, we decided that the database can be easily replaced with a document based NoSQL database, such as Firebase datastore [7]. Other advantages with Firebase datastore are low maintenance and matured integration with Android applications. Therefore, we decided to implement the analytic logics separately using python for its extensive use in data analysis and graph generation. Last, Paco requires participants to use their Google accounts to sign up, which is

contradictory to our object of user anonymity. With PrivaDroid, we only require the Google advertising ID [3] in order to associate the data with each participant. In result, we decided to develop PrivaDroid from scratch as our experiment instrument.

5.2 Frontend Android Application

The frontend of PrivaDroid is an Android application that supports Android SDK 23 (Android Marshmallow), which introduced runtime permission in 2015, through the latest Android SDK 29 (Android Q). Users can download the app either from Google Play Store or through other online app stores.

5.2.1 Choice of Participant Unique Identifier

Since the data collected should be associated with individual participants, we need an unique ID to distinguish between different users. To preserve user anonymity, we do not collect any personal account, name or personal identifiable information that can be easily linked to the user but rather some kind of identifier that is formal and every user possesses. We selected Google advertising ID [3] based on the guide provided by Android [1]. The reasons are twofold. First, we should not use any hardware identifiers such as MAC address and SSAID (Android ID) as user id because it requires privileged permission to access it in some Android versions. Second, despite that participants can reset their Google advertising IDs, we can cache them on users' devices using Android SharedPreferences API [4] and use those as users' ids for any subsequent data communication. As a result, we decided to use Google Advertising Id as the unique user identifier.

5.2.2 Application Description

This section explains in detail the application workflow, event detection, and app layout and other implementation details of PrivaDroid.

Application Workflow

This section describes the major workflow of PrivaDroid including the process of joining our experiment, survey prompting, and exit and reward survey answering.

Join Process. When participants open the app for the first time, PrivaDroid will detect the system language and country code on their phones. If they are not within the 4 languages and the 10 countries we target, PrivaDroid will warn them that they will not be qualified for the $10 USD compensation if they join. Users are then required to read through the consent form and explicitly agree to the terms before advancing to the setup screen. Figure 5.1 shows the consent form screen where participants need to read through the terms and check the agree checkbox to advance to setup instructions. The exact consent form in English is available in Appendix A.1. Once participants give their consent, they will arrive at a screen shown in Figure 5.2, a description of how to use PrivaDroid, and then Figure 5.3 and Figure 5.4, which instruct them to enable accessibility service and app usage access for PrivaDroid. The "SET UP" buttons in Figure 5.3 and Figure 5.4 take participants to the accessibility service and app usage access screens in Android Settings menu. Those who finish the setup can officially join the experiment. After users join, they are asked to answer the demographic survey. Figure 5.5 shows a

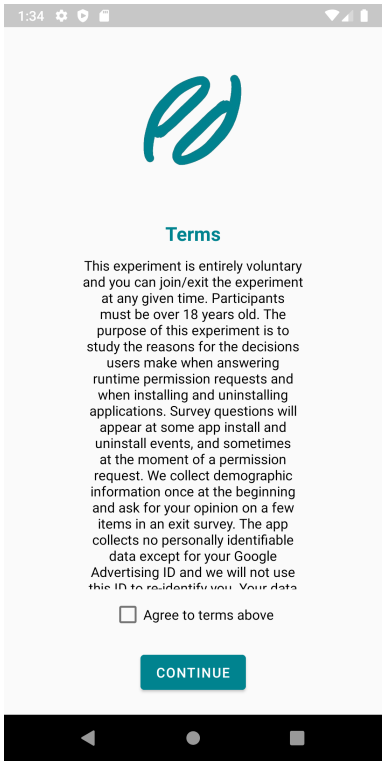

Figure 5.1: Consent form

screenshot of the demographic survey screen. Then they will arrive at the main screen, which consists of several tabs that can take them to screens of app install, app removal, permission and profile respectively.

Event Survey Prompt. After participants finish setup and demographic survey, PrivaDroid starts running in the background listening for app install and removal events and monitoring permission decision events. When events of one of the types happen, PrivaDroid will record the event and send it to Firebase datastore and create a customized notification to remind participants of answering the survey. For app install event, the notification text is set to "Why did you install <app name>?" where <app name> is replaced with the actual app name. Similarly, notification text for app removal events reads "Why did you uninstall <app name>?". For permission decision survey notification, we include the app name, permission name and the decision. The notification is structured as "Why did you grant/deny <permission name> to <app name>?" to refresh participant's memory. Figure 5.6, Figure 5.7 and Figure 5.8 show the survey notifications for an install event, a removal event as well as a permission deny event.

In order to not overwhelm participants with surveys, we required a minimum of 5 minutes between consecutive surveys. We achieved this by keeping track of the timestamp of last notification created and check if it has past the 5 minutes cooldown before creating a survey notification.

Exit Survey.

Participants are required to answer an exit survey at the end of the 30 day study to complete the experiment and receive the compensation. The questions in the exit survey measure privacy attitudes of the participant, which were derived from the well established IUIPC privacy scale [32]. We included questions along four dimensions, Control, Awareness, Collection and Secondary Use. As the IUIPC scale

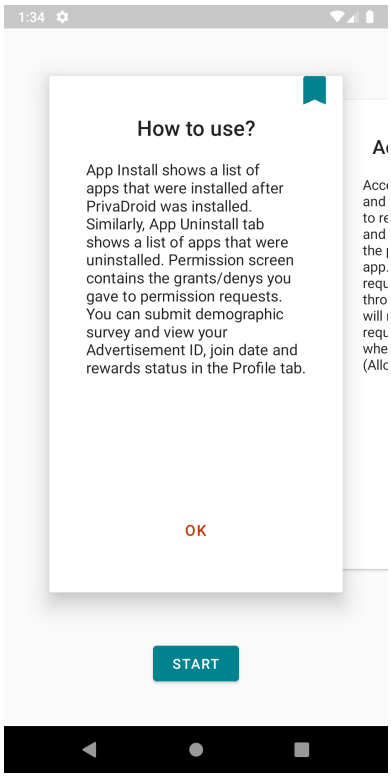

Figure 5.2: How to use PrivaDroid

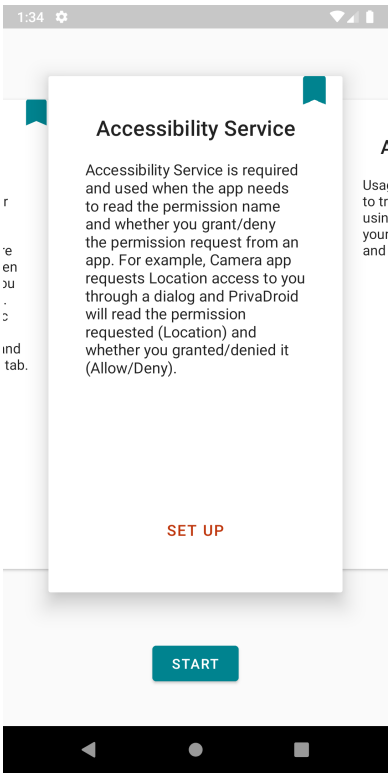

Figure 5.3: Enable accessibility service

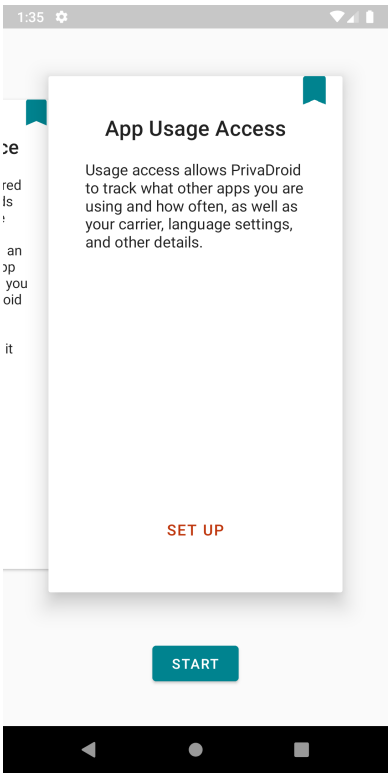

Figure 5.4: Enable app usage access

was originally developed in 2004 and focused on general "Internet use", we adapted the questions in a minor way to focus on mobile privacy. Specifically, we replaced the term "online companies" with "smartphone apps", and replaced the term "consumer online privacy" with "mobile app privacy". Our 15 questions (See Appendix A.2.6) were scored on a 5-point Likert scale, as opposed to the original 7-point scale as we learned that multilingual surveys are more frequently done with 5-point scales [46]. We mapped the answers to the range $\{-2, 2\}$. To evaluate the quality of our mobile-specific IUIPC questions, we conducted a 100 person Amazon Mechanical Turk survey and ensured that the survey questions are closed related to each other. We calculated the Cronbach's Alpha [13] coefficients, a measurement of internal consistence of a set of items, which are the mapped real values in our case. Cronbach's Alpha is calculated based on the number of items, the average inter-item covariance and the average variance. The resulting Cronbach's Alpha scores of our questions were in the range of 0.65 to 0.82, which met the minimum requirements. Both the PrivaDroid and mTurk surveys include a simple attention check question to ensure that participants are actually reading the questions, and we discard the data of participants who fail to correctly answer the question.

Reward Survey. After keeping PrivaDroid installed and running in background for 30 days and completing the demographic and exit survey, participants can submit their PayPal accounts in PrivaDroid for compensation.

Event Triggers

This section explains how PrivaDroid detects app install, removal events and permission decision events.

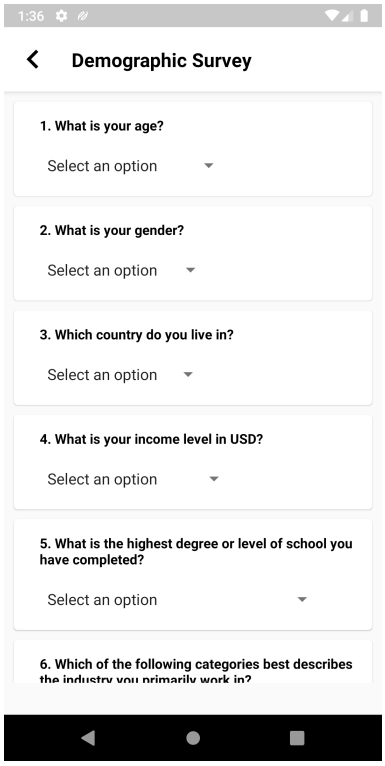

Figure 5.5: Demographic survey

Detecting App Install and Removal. When an Android APK is installed on a device, Android system will broadcast an `ACTION_PACKAGE_ADDED` system intent. Apps can register a broadcast receiver in the manifest in order to receive this *implicit* broadcast. An *implicit* broadcast to an app is a broadcast that is not targeted at the app specifically. Android 8.0 introduced a broadcast limitation which does not allow apps that target Android 8.0 or higher to register broadcast receivers in their manifests to receive *implicit* broadcasts. To circumvent this restriction and continue to listen for app install events on devices with Android 8.0 and above, we implemented a *foreground service* that would always run on participants' phones to register such broadcast receivers. When an `ACTION_PACKAGE_ADDED` intent is received, PrivaDroid records the app package name, app name and the version number and send them with some metadata in an event to the backend. In the meantime, PrivaDroid will create a notification which participants can click and answer the survey questions.

For app removal events, we used the same approach to detect them. When a package is removed, Android system broadcasts an *implicit* intent called `ACTION_PACKAGE_REMOVED`. Registering broadcast receiver for app removal intents suffers from the same restriction of limited system *implicit* broadcast reception. Therefore, we use one *foreground service* to register the receiver for both package install and removal events intent. Similarly, a notification for removal survey is created upon detecting a removal event.

Detecting Permission Decision. Capturing permission events is a bit more challenging as no system intent is broadcast when a permission request is granted or denied. A seemingly obvious way to observe permission changes is to consistently poll the permission granted for an app using the Android `getInstalledPackages()` API and passing the `GET_PERMISSIONS` flag. However, this approach can

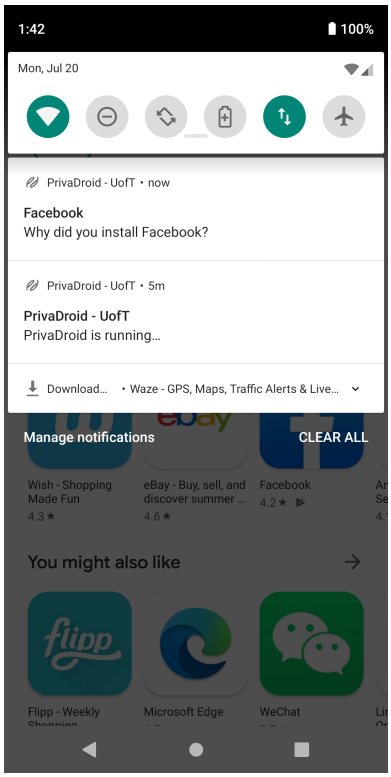

Figure 5.6: App install survey notification

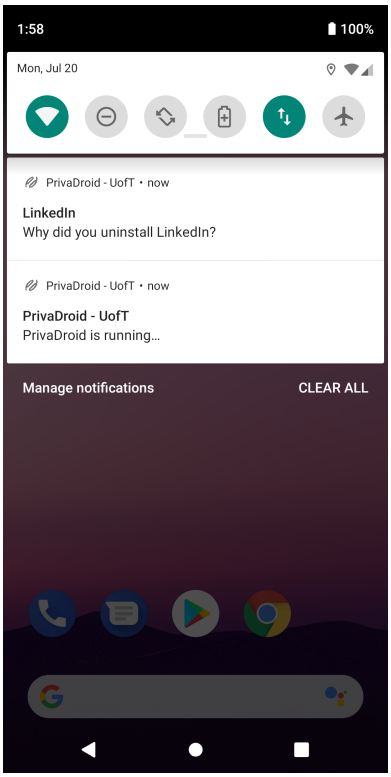

Figure 5.7: App removal survey notification

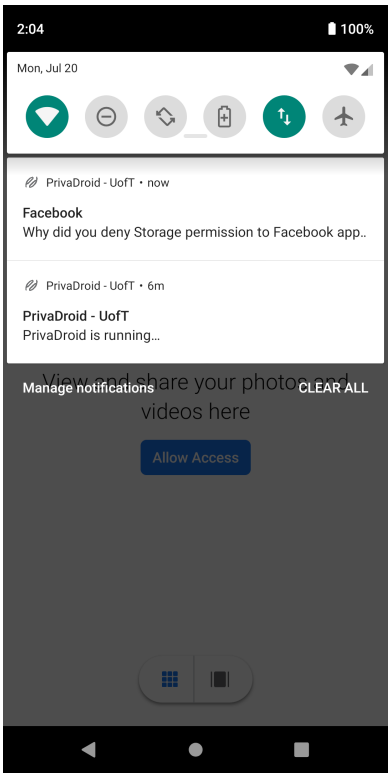

Figure 5.8: Permission deny survey notification

only capture permission changes but not the permission decisions that do not result in any change. For example, denying a request for a permission that was already denied before will not be caught. Instead, PrivaDroid implements an accessibility service facility to monitor the screen that participants view and look for UI elements with specific strings or View IDs to detect permission prompts, and uses the app usage permission to detect which app package requested the permission. Specifically, we look for deny button, whose View ID is always `com.android.packageinstaller:id/permission_deny_button`. Once we locate a permission request dialog, we extract the permission name by using regex comparison. This is made possible because the strings used in Android permission request always follow a similar structure with little variation between different permission types and languages. The structure of the English version is always `Allow` *app_name* `to` *permission_description*`?`, where *app_name* is replaced by the actual app name and *permission_description* by a short description of what the permission allows the app to do. For example, the system permission request for `Microphone` permission in English is `Allow Facebook to record audio?` App package requesting the permission can be extracted by querying the `UsageStatsManager` for the last active app. For other languages, we used the structure of the permission request as well as the *permission_description* provided in Android open-sourced code repositories [5, 12, 11]. After extracting the package name and permission type, we can then know which of the grant and deny buttons participants click by checking the button View ID in the accessibility event of type `AccessibilityEvent.TYPE_VIEW_CLICKED`.

Alternatively, Android users can toggle permissions in the Android Settings menu. Similar to runtime permission prompts, PrivaDroid uses an accessibility service to look for certain UI elements to track

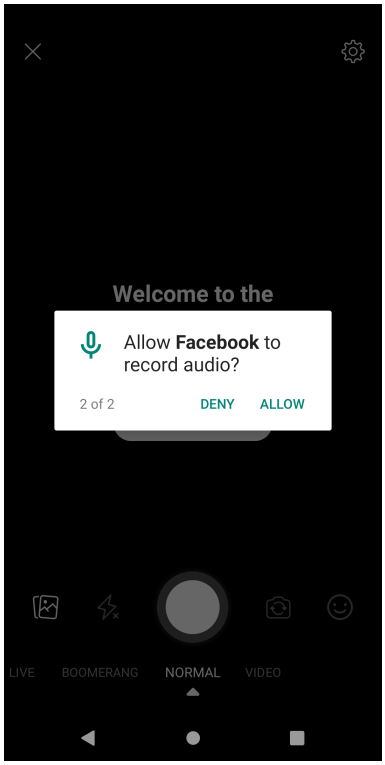

Figure 5.9: Example of runtime permission request structure

which permissions participants are toggling and which apps participants are doing it for. Based on the information, it then extracts the app name, permission name and the participant's grant/deny decisions. Figure 2.3 shows the two places where users can change their permission settings, which contain the app name as well as permission type users are changing. We can check whether the switch of the permission is on or off and thus know users' decisions.

PrivaDroid also records the time participants spent on reading and making decision on the runtime permission request prompts. This information is used to analyze whether the amount of time participants used affect their decisions and which permissions participants tend to spend more time on.

Detecting Permission Rationales. Some apps implement rationales explaining why a permission is needed. These rationales usually are presented in a dialog, along with buttons for users to grant or deny the rationale messages. Figure 5.10 shows an example of a permission rationale request Facebook provides in the app before making a system permission request of *Location*. Here, the user can select 'Allow', which triggers a subsequent system runtime permission request. Alternatively, the user can deny this permission rationale request and no system request will prompt. PrivaDroid was designed to capture these permission rationale requests and participants' decisions. Denying a rationale request has the side effect of reducing the number of permission requests made by an application via the system APIs, as well as artificially under-counting the number of permission denys that would be captured by PrivaDroid's monitoring of the permission decisions via the system APIs. To measure this effect, as well as measure the frequency of applications using such permission explanation dialogs, PrivaDroid captures the text on these dialogs using a keyword-based heuristics and the accompanying button that was clicked. This is achieved by implementing an accessibility service to look for texts and other UI elements on the screen

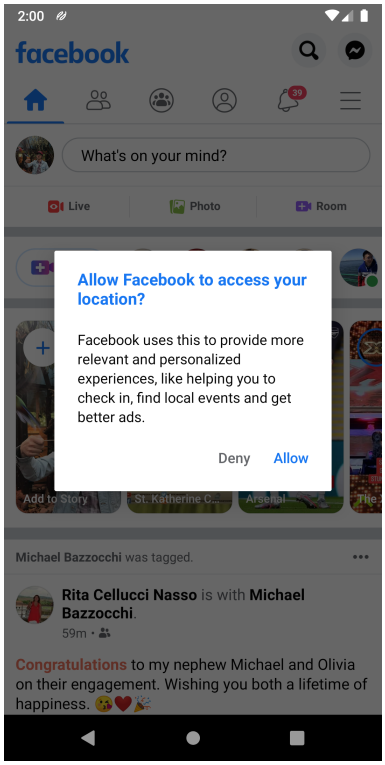

Figure 5.10: Example of *Location* permission rationale request in Facebook

that potentially explanation on why the app requires a permission. To qualify as a potential rationale, it must have one `TextView` or more that contain a word that is related to data collection and a word regarding a permission type and at least one `Button` allowing the user to grant or deny the message.

App Layout and Implementations

This section talks about each important screen of PrivaDroid and its implementation details.

App install, App removal and Permission. App install, app removal and permission screens share the same design pattern: an overview screen containing two card fragments, which include the number of surveyed/unsurveyed events and buttons that direct to a list of surveyed/unsurveyed events. Clicking on an item in the surveyed list takes the participants to a screen where they can answer and submit their responses. Similarly, clicking an item in the unsurveyed list directs to a screen where participants can view their read-only responses to the events. Figure 5.11, Figure 5.12 and Figure 5.13 demonstrate the permission survey overview, list of unsurveyed surveys and a deny permission event survey.

Profile. Profile tab contains the participant's information. It includes participant's unique id (Google advertising id), when he/she joined the experiment, demographic survey, exit survey and reward survey. Participants can check their responses to the demographic, exit and reward surveys through the corresponding accordion items shown in Figure 5.14 and Figure 5.15.

Participants need to keep the accessibility service and app usage access enabled during the length of the participation. In case that participants turned them off accidentally, we implemented a daily scheduled job using `JobService` [4] that checks whether these two settings are enabled and creates a notification to remind participants of enabling them if not enabled. When PrivaDroid checks these

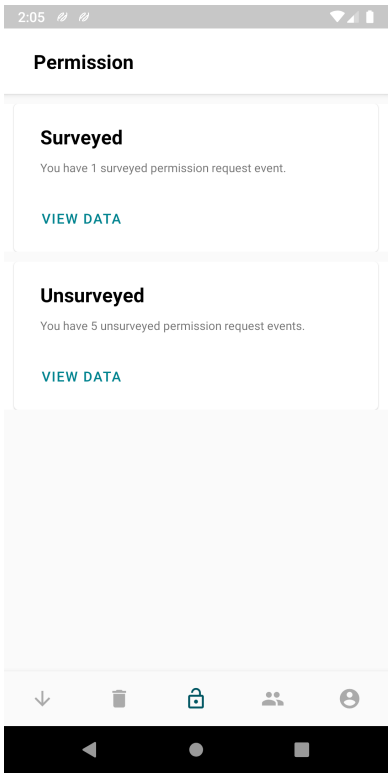

Figure 5.11: Permission survey overview

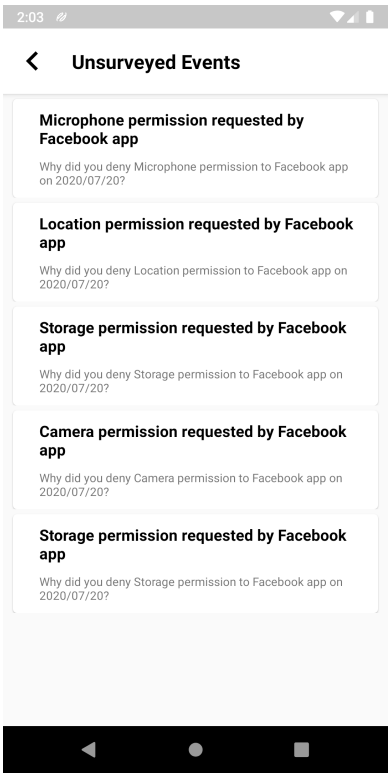

Figure 5.12: Unsurveyed permission surveys list

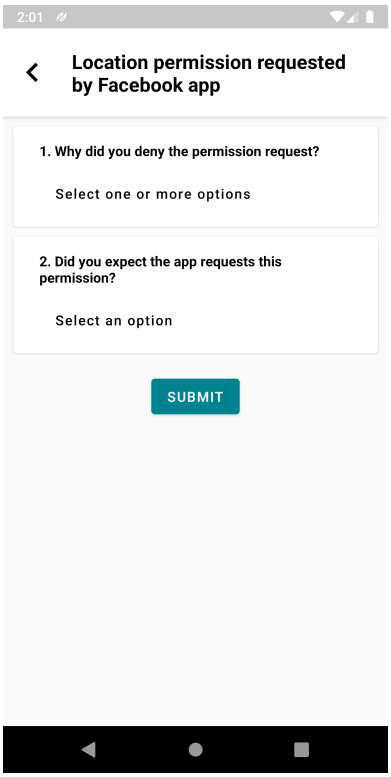

Figure 5.13: Answering a permission deny survey

two settings, it records the two boolean values representing their status and sends to the Firebase datastore as a daily heartbeat message used in validating participants' data and keeping track of the active participants in the experiment. The "Last Heartbeat Reminder" accordion shows the last time a daily heartbeat message was sent to Firebase.

5.2.3 App Localization

In order to include non-English speaking participants, we translated and localised PrivaDroid into Chinese (Traditional), Spanish and French. The translation consists of two parts: 1) strings in the PrivaDroid app, such as the consent form, the survey questions and answers, etc.; and 2) strings in the Android System UI, such as those used in detecting the permission changes participants made on the Android Settings page, Android system runtime permission dialogs and participants' decisions. For the first part, we used the translation service provided by the Google Play Console and had native speakers check the translations. For strings involved in the Android System UI, we used the translations provided in the open-sourced Android framework Git repositories.

5.3 Backend Infrastructure

PrivaDroid only requires a NoSQL database. We chose Firebase as the infrastructure for PrivaDroid because it provides a cloud database (Cloud Firestore) [7], application, user analytics (Analytics) and crash analytics (Crashlytics) [7] and is tailored for mobile applications.

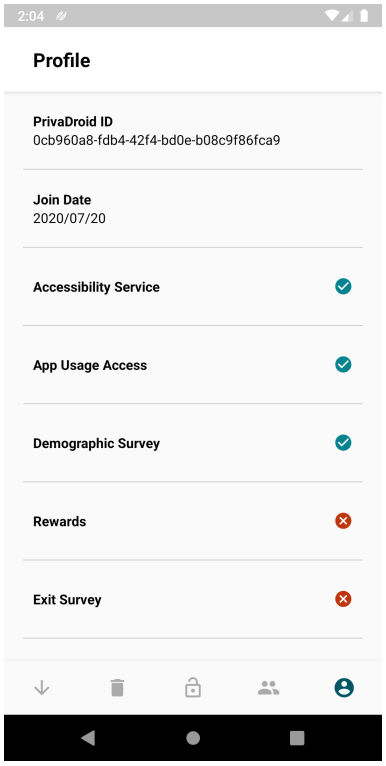

Figure 5.14: Top part of the Profile screen

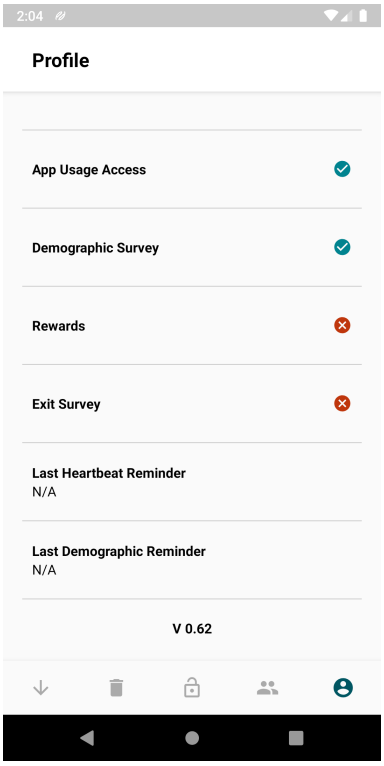

Figure 5.15: Bottom part of the Profile screen

5.3.1 Cloud Firestore

Cloud Firestore is a NoSQL document database that facilitates easy storing and querying data for mobile applications. It allows creation of document, which is a set of key-value pairs that contains the data. Documents are organized into collections. Each collection only contains (should contain) a single type of document. For our purpose, we created one collection for each significant event type that we want to record on participants' devices. These collections include but not limited to app install events, app uninstall events, permission decision events and etc. The details of each collection and the common and unique event fields are discussed below. We use the Google advertising id to differentiate the participants. We tagged each event sent from participants' phones to the Firebase datastore with their Google advertising ids and the timestamp of when the event happened.

Join Event Collection

When participants install our app and join our experiment, PrivaDroid sends a join event to Firestore. This event contains the Google advertising ID, which is used as the unique identifier of each participant, Android version of the device, carrier name, country code obtained from the telephony manager [4] (by accessing `getNetworkCountryIso()`), device locale, phone make and model and the timestamp of the join event. Device locale and country code were used to filter out the participants outside of our targeted 4 languages and 10 countries.

Demographic Survey Collection

Participants are required to answer a demographic surveys after they sign up. It asks about participants' age group, education level, country of residence, income, occupation, daily usage of phones and employment status. The answers will be used to categorize participants into each demographic bin in demographic analysis.

App Install Event Collection

App install event collection contains the information about an app installation event. Each document includes the app package (application ID), app name, app version and the Google advertising ID of the participant. It also contains a field called survey ID, which relates this app install event to an app install survey.

App Install Survey Collection

This collection contains participant's responses to app install survey questions. The questions and answers are stored in key-value pairs. The questions include, for example, what permissions the participant thinks the app requires. All questions are available in Appendix A.2.2. There is a field that contains the document ID of the corresponding app install event document in the app install collection.

App Uninstall Collection

App uninstall collection, similar to app install collection, contains documents of package uninstall events detected on users' devices. A document contains the same set of fields as an app install event.

App Uninstall Survey Collection

App uninstall survey collection hosts the survey responses to all app uninstall events. Individual survey asks why user uninstalls the app and the permissions that user remembered the app requested. The questions and answers are available in Appendix A.2.3. User advertising ID, the document ID of the corresponding app uninstall event and when the survey was completed are also stored each survey.

Permission Rationale Event Collection

In Section 5.2.2, we talked about the application of rationale messages provided by some apps before the system permission request. If the user grants the application's request, the application will subsequently call the Android API to create the actual permission request [2], which is prompted by the Android system; if user declines app's request, then the app will not ask the actual permission request. For these rationales, we captured the messages as well as participants' decisions on the buttons associated with the rationale (i.e. whether participants allow/deny app's rationale message). This collection contains the rationale, app name, app package and which button participants chose in those events. Each event has a uuid that will link a rationale event to its subsequent system runtime permission request event if these is any.

Permission Decision Collection

PrivaDroid detects permission decisions by users through run-time permission request dialog [2] or in the Android Settings menu. PrivaDroid captures the name and package of the app on which the permission decision happened, user's decision, i.e. whether user grants or denies the permission, any rationale message the app provides, any previous screen context text that will be used to understand the context in which the permission decision happened, and a survey ID that links the event to either a permission grant survey or permission deny survey survey document.

Permission Grant Survey Collection

A permission grant survey is presented to participants if they enable a permission in the settings screen or grant permission to a run-time permission dialog. The survey asks why users grant this permission, if they expected the app to request this permission and how comfortable they feel granting it. We also ask whether participants would prefer to only grant the permission temporarily. Although automatically depriving a permission does not exist in Android, we implemented a service that would remind the participants of toggling the previously granted permission if they preferred temporary grant. Permission grant survey questions and lists of options are provided in the Appendix A.2.4.

Permission Deny Survey Collection

Similar to permission grant survey, a deny survey is prompted if users deny a permission in app settings screen or a run-time permission dialog. Questions asked in the survey include why users denied the permission and whether they expected the user to request it. Questions included in the permission deny survey are available in the Appendix A.2.5.

Revoke Permission Reminder Collection

When participants would like to temporarily grant the permission and let PrivaDroid create a notification to remind them of depriving the permission access of an app in the permission grant survey, PrivaDroid will schedule a notification that will prompt after some time specified by the user. The collection contains events in which users click the notification and try to toggle the permission.

Exit Survey Collection

When users complete the experiment, they are asked to fill in an exit survey in PrivaDroid. The survey contains questions adopted and modified from IUIPC. In addition, we ask participants whether they are familiar with the Android permission system and let them select a list of permissions that they do not understand.

Reward Survey Collection

After users stay in the experiment for 30 days, they are eligible for a $10 reward through PayPal. Individual reward request consists of the user advertising ID, join and submission dates, and a PayPal account.

Heartbeat Collection

In order to calculate how many active users are in the experiment, we implemented a daily heartbeat event that gets sent to Firestore. Each event contains the user advertising ID, whether the accessibility service and app usage access are enabled since they are required for PrivaDroid to work correctly and the timestamp of the heartbeat.

5.3.2 Crashlytics

Crashlytics enables us to track and fix bugs and stability issues of the app as quickly as possible during the development of PrivaDroid. When a crash happens in PrivaDroid, Crashlytics captures the exception information such as the type of exception, the line number of the code producing the exception as well as the PrivaDroid version. In addition, it also provides the Android version and phone make of users' devices, which can be helpful in pinpointing the issue.

Chapter 6

Data Analysis

6.1 Data Summary

We conducted two studies, one to study the difference between the 10 countries across 5 continents and one to explore the behavioral and privacy attitude change of participants from the 5 English speaking countries (i.e. Canada, United States, United Kingdom, India and South Africa) under the influence of Covid-19 [6]. These two studies in conjunction ran from November 2019 to July 2020. For both studies, we advertised our study on the three advertising networks and initially targeted our ads towards females to encourage their participation. After reaching a sufficient number of female participants, we relaxed our targeting criteria and showed ads to all. In total, we spent $17,915.62 USD on advertising to recruit the participants, which generated 4,189,911 impressions, 44,112 clicks and 8,019 installs of the PrivaDroid app. Of the participants who installed PrivaDroid, 2,429 participants stayed for the required 30 days period to complete the study. Among them, 1,435 identified themselves as males, 964 as females and the rest identified as neither or preferred not to disclose their gender. Another 1,784 participants join the experiment but withdrew before the required period, and their data is not included in our analysis as a result. (Based on the number of app installs the advertising generated, many participants downloaded the app but did not join the study.) For the the first study, the advertising was carried out in two steps. We advertised to recruit participants from the 5 English speaking countries and all participants finished by February 2020, which was before the Covid-19 breakout. After they finished, we started recruiting participants from the other 5 countries (i.e. Spain, Argentina, France, Singapore and Hong Kong) and all participants finished by June 2020. For the second study, we started the second round of advertising for the 5 English speaking countries in March 2020 after Covid-19 broke out. We reached our target, which is at least 50 male and female participants for each country, and all participants completed the 30 days requirement by July 2020. This chapter focuses on analysis of the data of the first study and Chapter 7 discusses our findings on impact of Covid-19.

For the first study, Hong Kong was the only region where we did not reach our aim of 50 female participants; thus we only use the Hong Kong data for aggregate analysis hereafter, however not for demographic analysis. Table 6.1 summarizes the breakdown of the participant counts across the 10 countries and regions. During the study period, as shown in Table 6.2, our participants saw 74,381 app install events of which 36.2% were surveyed, 67,094 app removal events of which 28.6% were surveyed and 36,095 permission events of which 29.7% were surveyed. Note that due to our self-enforced limitation on

Country and Region	Males	Females	Other	Prefer not to say
United States	98	131	3	2
Canada	108	75	5	1
United Kingdom	85	55	0	0
India	202	60	0	0
South Africa	55	70	0	0
Singapore	60	54	0	1
Spain	128	91	1	4
Argentina	190	62	0	1
France	98	56	1	0
Hong Kong	55	27	0	1
Total	1,079	681	10	10

Table 6.1: Country and Gender Demographics for Non-Covid Analysis

Event Type	Total # of Events	Total # of Surveys	Avg # of Events	Avg # of Surveys
App Install	74,381	26,926	41	15
App Removal	67,094	24,288	37	11
Permission Decision	36,095	13,066	20	6

Table 6.2: Data Overview of App Install, App Removal and Permission Event

how frequently surveys were prompted to participants, not all events resulted in a survey being triggered.

6.1.1 App Install

In total, we observed 74,381 app install events, that is 41.2 events per participant on average. 36.2% of the install events were surveyed, which results in 14.9 install surveys per participant.

We ask why participants decide to install an app right after installation. We list the reasons for app install and the frequency in which they were selected in Table 6.3. The most popular reasons participants selected for installing an application are "I want to try it out" (33% of the surveys), "The app is useful" (23% of the surveys), and "The app is cool or fun to use" (15% of the surveys). This indicates that users cared about whether the app can be useful or fun when deciding to install an app. The first two reasons remain the same as those reported in [16] but with a decrease of 16% and 7%. "The app is cool or fun to use" ranked the fourth at 26% in [16]. However, it surpassed "The app is part of a product/service I use" by 5% to the third place in our study. Participants selected "The app has fewer permissions than other apps like it" 4% of the times, which doubled from 2% reported in [16]. This suggests that people more often take the permissions an app requires into consideration when choosing apps.

6.1.2 App Removal

During the study period, these participants saw 67,094 app removal events, of which 28.6% were surveyed. Each participant averaged 37.2 events and 10.6 surveys.

Similarly, we survey the participants immediately after they remove an app. The app removal reasons and number of times each reason was selected are listed in Table 6.4. The top three most selected reasons for uninstalling an application are "I no longer use the app" (37% of the surveys), "To free up space or speed up my device" (26% of the surveys), and "I didn't like the app" (22% of the surveys). The choices of uninstall reasons we observed are exactly same as the ones reported in [16]. "The app required

Install Reason	# (%) of Surveys
I want to try it out	9,102 (33%)
The app is useful	6,166 (23%)
The app is cool or fun to use	4,075 (15%)
I was offered something in return (e.g. credits, monetary rewards, discount)	3,900 (14%)
I was required to install it	2,913 (11%)
The app is part of a product/service I use	2,897 (10%)
I trust the app or the company making the app	2,774 (10%)
My friends/family use it	2,658 (10%)
It was the only app of its kind (no other apps provide the same functionality)	2,187 (8%)
The app has fewer permissions than other apps like it	1,199 (4%)
Other	1,041 (4%)
None	712 (3%)

Table 6.3: App Install Reasons

Removal Reason	# (%) of Surveys
I no longer use the app	7,067 (37%)
To free up space or speed up my device	5,010 (26%)
I didn't like the app	4,317 (22%)
The app is not working as expected	3,177 (17%)
The app is crashing/very slow	1,933 (10%)
Because of advertisements in the app	1,340 (7%)
Other	1,336 (7%)
Because of in-app purchases	1,004 (5%)
The app required permissions I wasn't comfortable granting	974 (5%)
None	494 (3%)

Table 6.4: App Removal Reasons

permissions I wasn't comfortable granting" was only selected in 5% of the removal surveys, which is similar to the 4% reported in 2017. This indicates that users' willingness to grant a permission is not a primary reason of their decisions to uninstall an app.

6.1.3 Permission

Permission Denials

Of the 36,095 permission decision events across the 11 permission groups, we found that the participants denied an overall of 16.6% of the permission requests. Even without considering the events for recently introduced permissions (such as *Body Sensors*, *Call Logs* and *Physical Activity*), the aggregate average deny rate is close to the 16% reported in an earlier study [16]. Based on the data observed in our current study, 8.8% of the permission decisions occurred from the Android Settings menu, which is similar to the 5% reported in [16]. For these two aggregate metrics, the behavior has not changed much since 2017. Among all the permission decisions participants made via the Settings menu, 39.9% were to deny a previously granted permission. While this number if high, it still suggests that the majority of decisions that happened in the Settings menu are to grant a permission. As we will see later, a top reason for

Permission Event Type	# of Events	Overall Deny Rate
Runtime Prompts	32,919	14.4%
Android Settings	3,176	39.9%

Table 6.5: Deny Rates of Permission Events by Runtime Prompts and Android Settings Menu

Permission Type	# (%) of Events
Storage	9,827 (27.2%)
Location	9,223 (25.6%)
Camera	5,534 (15.3%)
Microphone	3,889 (10.8%)
Contacts	3,318 (9.2%)
Phone	2,968 (8.2%)
SMS	755 (2.1%)
Calendar	374 (1.0%)
Call Logs	138 (0.4%)
Physical Activity	36 (0.1%)
Body Sensors	32 (0.1%)

Table 6.6: Number (Frequency) of permission requests for individual permission types

denying a permissions is because participants are aware that they can go to the Settings menu and change their decisions afterwards.

Both the number of events and deny rates vary a lot based on the individual permission type. Table 6.6 shows the number and frequency of permission decision events for each permission type. While *Storage*, *Location*, and *Camera* are prominently requested with each having $>$ 5K events, we see very few permission decision events for *Body Sensors*, *Call Logs*, and *Physical Activity* permissions. This phenomenon could be because that these three permissions are fairly new.

Figure 6.1 shows that deny rates for each permission group with at lease 50 decision events in our experiment (*Body Sensors* and *Physical Activity* permission groups are omitted). *Microphone*, *Calendar* and *Contacts* have the highest deny rates of 30.0%, 24.2% and 18.9% respectively. Permissions including *Location* and *Storage*, which are also the most frequently requested in our data, have much lower deny rates of 15.5% and 12.0%. Note that the overall average permission deny rate of 16.6% observed across all permissions was computed by summing the total observed grant and deny decisions, but not based on the average of deny rates for individual permission types. Compared to deny rates recorded in [16] (which only had US participants), we observe that deny rates for US in our data have increased for *Calendar* (21.7% from 10%) and *SMS* (16.4% from 10%), and decreased for *Phone* (12.8% from 19%), *Location* (8.5% from 15%), and *Camera* (11.1% from 15%). We discuss these differences with [16] further in Section 6.1.3.

Approximately 11% of our participants were using Android 10 devices, and thus had access to the *foreground only* permission option introduced in it for *Location*. As shown in the Table 6.7, although deny rate for the *Location* permission on Android 10 and other versions were about the same at 16.5% and 15.3% respectively, two thirds of the *Location* permission grants in Android 10 were *foreground only*. This indicates that the *foreground only* option is heavily used by our Android 10 participants. Since the option is only available for *Location* permission and in Android 10 alone, which did not make up a big portion of the collected data, we treat *foreground only* option as a permission grant in our analysis.

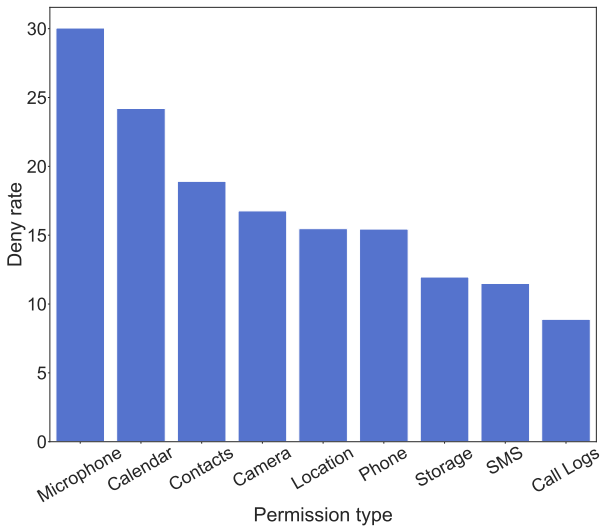

Figure 6.1: Permission deny rate of each permission type

Android Version	# (%) of Grants	# (%) of Denys	# (%) of Foreground Onlys
Android 10	423 (29.1%)	238 (16.5%)	788 (54.4%)
Other Versions	6,578 (84.7%)	1,196 (15.3%)	0 (0%)

Table 6.7: Number of grants, denys and foreground onlys of *Location* permission for Android 10 and other versions

In examining the rationales our participants gave for denying permission, we discover that the top three reasons for denies are: "I can always grant it afterwards if I change my mind" (26% of denies), "I do not use the specific feature associated with the permission" (25% of denies), and "I think the app shouldn't need this permission" (23% of denies). The first two reasons indicate that participants are aware that they have the ability to revise their permission grant and deny reasons, and that they try to enforce the principle of least privilege by denying permissions that they believe aren't required for the app features that they use. The third reason indicates either that users do not see a clear reason why a permission should be needed, or that an app may indeed bu asking for unnecessary permissions. Overall we see participants considering functionality as a primary aspect of decision making. The third reason provide further motivation for apps to provide explanations. Compared to [16], the top reasons have mostly remained the same. All of the permission deny reasons and how often they were selected are shown in Table 6.8.

Similarly, the top reasons for permission grants shown in Table 6.9 include: "I want to use a feature that needs this permission" (37% of grants), "I think the app won't work otherwise" (25% of grants) and "I trust the developer" (23% of grants). These top reasons are consistent with the reasons indicated in [16], and trust in developer still seems to play an important role in whether participants decide to grant a permission to an app. Overall, the top reasons for both grants and denies suggest that participants tended to rationalize their permission granting and denying as a trade-off between functionality and privacy - reasons that suggest a more emotional response, such as "I have nothing to hide" or "I wanted the permission screen to go away" were chosen less often.

We logged the amount of time participants spent reading the permission dialog before making a

Permission Deny Reason	# (%) of Surveys
I can always grant it afterwards if I change my mind	463 (26.5%)
I do not use the specific feature associated with the permission	433 (24.7%)
I think the app shouldn't need this permission	397 (22.7%)
I wanted the permission screen to go away	274 (15.6%)
I consider the permission to be very sensitive	268 (15.3%)
I don't trust the developer	201 (11.5%)
The app gave a poor explanation	147 (8.4%)
I think something bad might happen	136 (7.8%)
Other	127 (4.1%)
None	113 (2.6%)

Table 6.8: Overall Permission Deny Reasons

Permission Grant Reason	# (%) of Surveys
I want to use a feature that needs this permission	3,312 (36.8%)
I think the app won't work otherwise	2,275 (25.2%)
I trust the developer	2,028 (22.5%)
The app gave an explanation that made sense	1,381 (15.3%)
I have nothing to hide	1,355 (15.0%)
Because the app is popular	1,007 (11.2%)
I want the permission screen to go away	992 (11.0%)
The developer already has this information about me	732 (8.1%)
Other	369 (4.1%)
None	231 (2.6%)

Table 6.9: Overall Permission Grant Reasons

decision. We discovered that the average dialog read time for granted events was 3.1 seconds, whereas participants spent 4.4 seconds before denying a request. This suggests that our participants were more reluctant when it came to denying a permission request.

Temporary permissions. We also asked participants each time after they granted a permission, if they would have liked to grant it temporarily. We found that 24% of the times participants chose to grant a permission, they would have preferred to do so temporarily. Among the permissions that were surveyed at least 50 times, the desire to grant temporarily ranged from 21% to 26% depending upon the permission. For example, we see *Microphone* at 26%, *Location* at 25% and *Camera* at 24%. In line with this, the upcoming Android 11 OS release [41] includes a *one-time* grant option for *Location*, *Microphone* and *Camera* permissions. One could interpret the desire to grant temporarily as a hesitation, or lack of comfort, in granting a permission permanently.

We compared how comfortable participants felt when granting permissions with their desire to grant permissions temporarily. In the cases when participants indicated they were not interested in granting a permission temporarily, 53% of them selected that they felt either very or somewhat comfortable granting those permissions. However among those who said they would have liked to grant the permission temporarily, only 36% of them felt very or somewhat comfortable. We list the fraction of surveys in which participants felt comfortable granting app the permission when they did or did not desire to grant the permission temporary in Table 6.10. This decrease in percentage of comfortable permission grants is observed across all permission types when participants preferred a temporary grant: for *Calendar* we

Permission Type	Temporary Grant	Non-Temporary Grant
Calendar	30.0%	55.4%
Camera	34.4%	56.6%
Contacts	27.0%	50.4%
Location	32.0%	52.8%
Microphone	41.6%	52.2%
Phone	30.5%	42.3%
SMS	40.0%	46.1%
Storage	35.0%	51.1%
Overall	36.2%	53.1%

Table 6.10: Fraction of surveys participants felt comfortable granting the permission when they did or did not desire temporary grant

see the largest decrease of 25% and the smallest decrease is 6% for *SMS*. This indicates that one reason why users would like to grant permissions temporarily is due to comfort.

Explanations

It is important for any app to provide an explanation or context to the user before requesting a permission [35, 25, 21, 30]. This context could be implicit, for example, a well designed app flow; or it could be explicit, such as a message explaining the reason for the permission request. As mentioned earlier in Section 5.2.2, these permission explanation messages can be text dialogs shown by the app with some UI elements (such as buttons) for users to interact with. PrivaDroid captures these explanations by scanning for Android TextViews that occur right before a permission request, and capturing those that contain a verb that is related to data collection and a noun that belongs to a permission. We then associate this explanation message with the respective permission request. We also record the button options present on the dialogs and what was clicked by the study participant (to determine if the participant approved/denied the request). We acknowledge that this heuristic is incomplete and may miss some explanations (i.e. when images are used instead of text), but nonetheless did capture a many explanation messages during our study.

In total, we collected 1648 permission explanation messages that preceded a grant or a deny across 1057 apps. Thus 15% of apps in our study include explanation messages for their permission requests. Having an explanation reduced the permission deny rates to 7.4% as compared to the 17.5% deny rate for requests with no explanations. Providing explanations helped participants make permission decisions faster. On average, participants spent 2.2 seconds before granting or denying when given an explanation, but spent 3.4 seconds on average for requests without an explanation. In summary, permission explanations made a significant difference for our participants, reducing the deny rate by more than 1/2 and shortening decision time by 1/3.

It is important to note that an explanation message dialog may cause a runtime permission request to be omitted. For instance, an app might indicate that it would like to "Use Location to show personalized ads?" with two buttons: "Not Now" and "Yes". Clicking on "Not Now" conveys to the app that the user is going to deny the permission request, so the app may simply skip making the request. Because PrivaDroid computes deny rates based on Android system permission requests, PrivaDroid will undercount these app-specific permission deny events. To adjust for this, each of the 2643 English explanation messages where a "Not Now" or an equivalent option was selected by the participant was manually

evaluated by two of the authors to determine if it is indeed a permission rationale message, resulting in 540 actual pro-active deny messages[1]. Counting these as denies results in an adjusted permission deny rate of 16.9% compared to a base 14.4% deny rate for English speaking countries (i.e. Canada, UK, USA, India and South Africa). Because this behavior only affected 15% of applications seen in our study, we use unadjusted deny rates in the remainder of the paper.

Expectations Versus Behaviors

Prior research has suggested that participants are more likely to grant a permission if they expect an app to ask for it [47]. It is typically hard to measure both expectations and behaviors "in-the-wild"—on participants' personal devices. PrivaDroid allows us to do so. We collect participants' expectations at two points in their journey to a permission request. The first one is during app install, when participants are asked "which of the following permissions do you think the app requires?" and they select as many as they want from the full list of permission groups. (See Appendix question A.2.2.) This captures whether a participant expects an app will ask for a permission before any permission requests occur. The second moment is after the participant has responded to a runtime permission prompt; they are asked "did you expect the app to request this permission?" (regardless of whether the participant granted or denied the permission). For this question, participants select either "Yes" or "No". Since participants have, at this point, already engaged with the app, and the permission request has already occurred, this measure captures whether a participant felt the request was expected or not.

Install-Time Expectations. The participants' expectations at install-time about a permission request may or may not be correct. We use the term *correctly expected* to refer to cases when the participant expected a particular permission would be requested and the app requested it, the term *incorrectly expected* captures cases when a participant expected a permission but the app did not request it, and the term *unexpected* to refer to cases when a participant did not expect the permission, but the app actually requested it.

We begin by examining whether our participants' install-time expectations match reality. Figure 6.2 shows rates for the three types of expectations for the 6 permission types with the most permission request events. Our participants' ability to correctly predict whether an application will request a permission is poor, ranging from 7% for the *Phone* permission to 19% for the *Location* permission. Moreover, more than two thirds of their expectations are wrong, in that they think a permission will be asked for that isn't.

We hypothesize that this behavior might come from participants becoming habituated to assuming that apps frequently request unnecessary permissions [17, 29, 44, 45]. Overall, this suggests that there still exists a significant gap both in the way applications set the expectations for users, as well as the user's understanding of how applications use permissions.

Figure 6.3 shows the deny rate for correctly expected and unexpected permission requests for individual permissions. (Note we cannot compute deny rate for incorrectly expected permissions since the app doesn't ask for the permission.) Interestingly, the deny rates are always higher when the permission request is unexpected, for each permission group. The average deny rate for expected permissions of 9.9%, and a deny rate for unexpected permissions of 14.1%. Hence the phenomenon of participants denying unexpected permissions more frequently holds in aggregate and across permission types. We

[1]Some of the explanations were actually permission requests by web pages in a browser

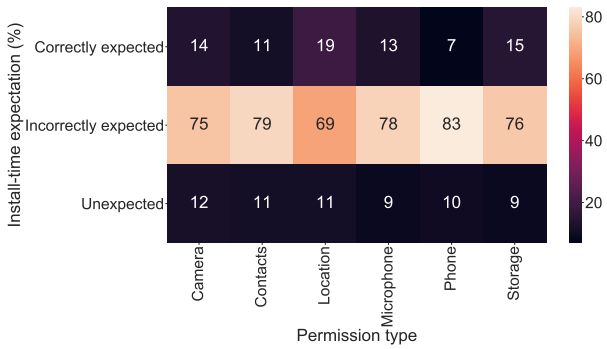

Figure 6.2: Permission expectations vs reality

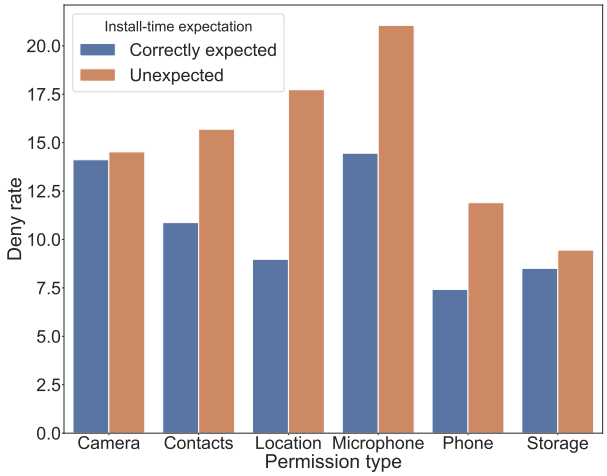

Figure 6.3: Permission deny rates for permissions expected/unexpected at install-time, by permission type

also used Pearson's Chi-squared test to verify the dependence between install-time expectation and the deny rate and found that $\widetilde{\chi}^2 = 28.2$, p-value = 1.072e-7 with df = 1. This affirms that participants were more likely to grant a request they predicted at install-time.

Runtime Expectations. In 7,750 (72%) of our surveyed runtime permission events, participants expected the permission request and in the remaining 2980 of our runtime permission surveys they did not. The number of permission events where an initially unexpected install-time permission request changed to an expected request at runtime (over all permission events where we recorded both install-time and runtime expectations) was 24% (1,239/5,111) demonstrating that users sometimes revise their expectations as they use and interact with an app. The deny rate for permissions expected at runtime was 12.2% whereas the deny rate for runtime unexpected permission requests was 26.8%. This ∼15% difference in deny rates is significantly larger than the ∼3% discrepancy observed for install-time expectations—participants are 2× more likely to deny permission requests they did not expect at runtime. Figure 6.4 shows that the denial rates for unexpected permission requests are roughly double that of expected requests, *across all the permission types*. In the case of the Phone permission, the deny rate tripled, going from 9% to 27%. The ensemble of these observations shows that expectations do influence participant behavior, and also provides a strong indication that with improved understanding and expectations around permissions, participants are more likely to grant a permission, across all permission types. We

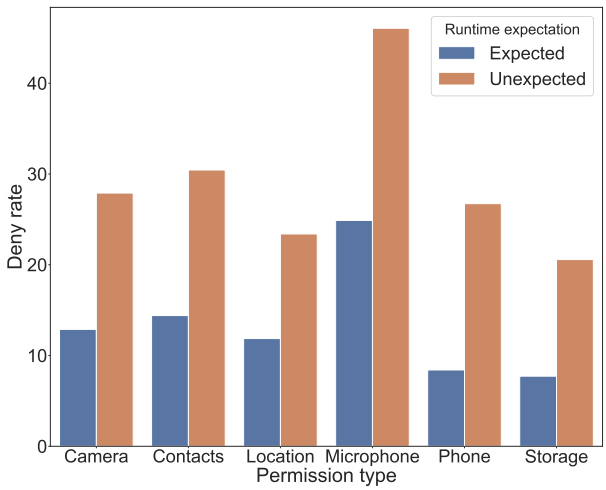

Figure 6.4: Permission deny rates, for expected/unexpectedpermissions requests at runtime, by permission type

computed Chi-squared value for the runtime and discovered that the deny rate is more dependent on runtime ($\widetilde{\chi}^2 = 328.7$, df = 1, p-value = 1.845e-73) than install-time expectation. Our findings corroborate the findings in [47], although as pointed out in Section 3, our study mechanisms are quite different and our study size here is two orders of magnitude larger.

Demographic Analysis

We now look at behaviors according to demographics. Our cross country comparison includes 9 countries (recall we leave Hong Kong out here since we were unable to recruit at least 50 female participants). Prior research has stipulated that user attitudes and behaviors around privacy differ across countries, perhaps due to cultural reasons, legal frameworks, etc. Table 6.11 shows the deny rates across different countries. The aggregate deny rate per country varies from 12% for the United States to 25% for Argentina. It is noteworthy that some regions (Argentina and Spain) have deny rates that were twice as high as other regions (the US and India). However this aggregate deny rate hides variation among participants within countries. We thus calculate the standard deviation of intra- and inter-country permission deny rates, also shown in Table 6.11. We find that the intra-country standard deviation is significantly larger than the inter-country standard deviation. This finding suggests that behaviors with privacy controls vary significantly within individual countries and this may overshadow an ability to compare countries with average metrics. The average number of permission events varies across countries (see Table 6.11), with India having the highest average of permission events at 34.1, which is almost triple the lowest value of 12.1 from participants in Argentina. Participants in some countries are more actively using apps and changing their permissions compared to others.

We examined the top reasons for granting and denying permissions across countries and found that the top 3 to 4 reasons for each country are essentially the same across all countries, indicating that participants across different countries make the same previously observed functionality vs. privacy trade-offs when deciding whether to grant a permission or not.

We next consider behavior differences with deny rates across gender. Table 6.12 shows the number of participants for each gender. As shown in Table 6.11, the deny rate of female participants is higher

Country and Region	Avg # of Grants	Avg # of Denys	Avg Deny Rate	Intra-country Deny Rate Std Deviation	Avg Privacy Sensitivity
US	27.1	3.7	12.1%	15.7%	1.10
Canada	15.4	3.4	18.2%	20.7%	1.26
UK	16.2	3.1	16.0%	20.6%	1.13
India	29.6	4.5	13.3%	15.8%	1.18
South Africa	15.9	2.6	14.0%	20.1%	1.39
Singapore	12.6	2.4	16.0%	24.1%	1.32
Spain	12.0	3.8	24.0%	24.2%	1.16
Argentina	9.1	3.0	24.8%	27.1%	1.20
France	11.5	2.6	18.7%	18.8%	0.98
Hong Kong	4.9	2.4	33.2%	33.8%	1.14
Overall	16.7	3.3	16.6%	6.0%[2]	1.18
Gender	Avg # of Grants	Avg # of Denys	Avg Deny Rate	Intra-gender Deny Rate Std Deviation	Avg Privacy Sensitivity
Male	17.7	3.3	15.9%	18.7%	1.14
Female	15.1	3.3	17.8%	19.4%	1.25
Other	23.2	3.6	13.4%	21.7%	1.30
Did not say	11.8	5.2	30.1%	27.7%	1.00
Education level	Avg # of Grants	Avg # of Denys	Avg Deny Rate	Intra-education-level Deny Rate Std Deviation	Avg Privacy Sensitivity
Less than high school	13.7	2.3	14.6%	19.6%	1.09
High school	17.2	3.0	15.0%	17.3%	1.18
Bachelor's or more	16.4	3.9	19.2%	21.0%	1.20
Did not say	18.9	4.7	19.9%	21.3%	1.07

Table 6.11: Permission Request Events and Decisions

than that of male participants by approximately 2% at 17.8% and 15.9% respectively. This difference is significantly smaller than the 2017 results in [16] where female participants denied nearly twice as often as male participants (20% for females and 11% for males). A key difference is that the prior study included US participants only.

We thus examined our US participants and found that the deny rate has actually decreased to 9.3% for female participants and increased to 15.2% for male participants. A primary reason for this difference could be due to the different participant recruitment methodologies adopted (advertising vs. company participant database). In addition, we also looked into the education level of the US participants in our study, and discovered that 86% of the females have a high school diploma or lower compared to 67% for males. Also, the deny rate of US males with a Bachelor's degrees is 32%, and this is 4 times higher than that of females with a Bachelor's degree. While we do not know the education level breakdown of males and females in [16], the variation in the education levels of US male and female participants in our study could also be contributing to the differences in the observed deny rates.

We also calculated the top reasons for granting/denying permissions for female and male participants. While the top reasons for granting a permission are the same, we observed some differences in the reasons for denying a permission. The top reasons for permission deny ("I do not use the specific feature associated with the permission" and "I think the app shouldn't need this permission") show up in 30% and 27% of denies for female participants, but they drop to 20% and 21% for male participants

Gender	# of Participants
Male	1,079
Female	681
Other	10
Did not say	10

Table 6.12: Number of participants of each gender

Education Level	# of Participants
Less than high school	152
High school	989
Bachelor's or more	573
Did not say	75

Table 6.13: Number of participants of each education level

respectively. This suggests that female participants may be a bit more focused than our male participants on how they use an app when considering permission decisions.

Finally, we examine how the deny rates for individual permissions vary across countries, shown in Figure 6.5. While *Microphone* is the most frequently denied permission overall, it has the highest deny rate in only 5 countries. *Calendar* is the top denied permission in 3 countries. This explains the overall high deny rates for *Microphone* and *Calendar*. Meanwhile, *Location* has the highest deny rate only in Spain. These variations in the top denied permissions across the different countries and regions studied indicates that permission sensitivity is not the same in every country, and even within a single country certain permissions are much more aggressively denied than others (e.g. deny rate for *Calendar* compared to other permissions in France).

Attitudes versus Behaviors

Each participant in our study was required to answer an exit survey that measured their privacy attitudes along the 4 dimensions of Control, Awareness, Collection and Secondary use of private information, as described in Section 5.2.2. Based on their responses to these questions, participants are assigned a score on a scale between $\{-2, 2\}$ in each dimension, with positive scores indicating higher sensitivity to privacy loss in that dimension. Also, we average out these dimensional scores, and assign an *overall privacy score*

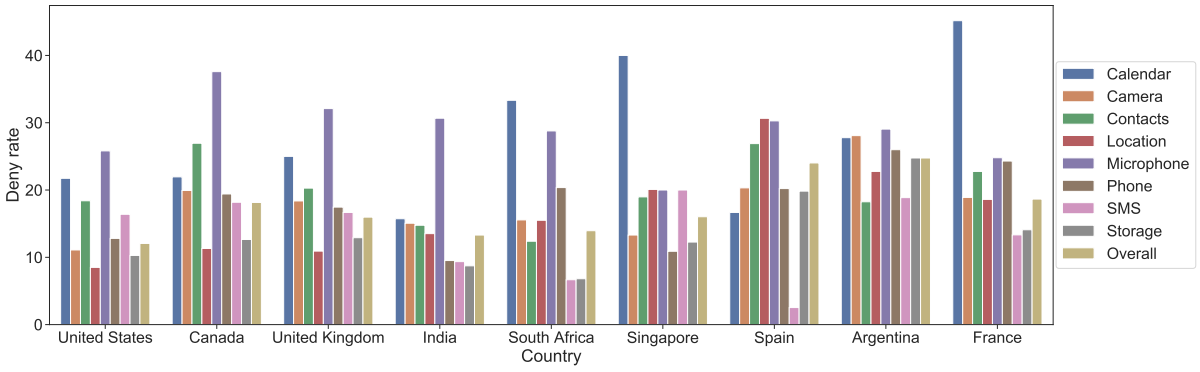

Figure 6.5: Permission deny rates of individual permission types in each country

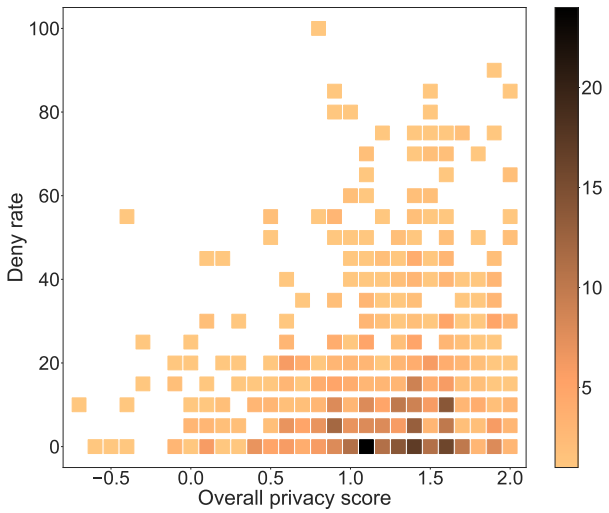

Figure 6.6: Scatter plot of participants >= 10 permission events by Deny Rate vs Privacy Sensitivity

to each participant. This overall privacy score would indicate the privacy sensitivity of the user. The responses of participants who failed the attention check question were not included.

To understand the relationship between their privacy attitude and permission deny behavior, we plot the scatter plot of participants who had over 10 permission events by their deny rate and privacy sensitivity in Figure 6.6. Each square in Figure 6.6 represents the number of participants who had the overall deny rate and privacy score specified by their locations on the x and y axes. The color bar on the right indicates the relationship between the color of a square and the number of participants. For example, the square in the bottom right corner means that there are 3 participants who had an overall privacy sensitivity of 2.0 and a deny rate of 0% (meaning they granted all the permission). From this, we make three observations. First, as expected, as overall privacy sensitivity increases, so does the average deny rate, with an increasing number of participants having a deny rate greater than the mean (16.6%). Second, the variance of permission denying behavior increases as overall permission sensitivity increases, with high variability of deny rates for participants with high sensitivity. Finally, and most interestingly, the distribution of deny rates for participants with relatively high overall privacy sensitivity (> 1.0) is not uniform—a large proportion, 25% (288/1176), have deny rates lower than the population average of 16.6% and show up as a concentration of users near the bottom middle right. This behavior, where participants indicate they are sensitive to privacy, but do not behave in a way consistent with that sensitivity has often been called the "privacy paradox" [26, 43]. While there has been no strict definition of what characteristics determine whether an individual exhibits the privacy paradox or not, for the purposes of this paper, we define the aforementioned 25% of participants in our study the "paradoxical group".

To better understand this paradoxical behavior we performed two analyses. First, we hypothesize that overall permission sensitivity may not be wholly indicative of the willingness of a participant to share information or have it collected on them. Thus, we use K-means to group the participants according to their scores in each dimension of Control, Awareness, Collection and Secondary use of private information. We set the number of groups to 4 after evaluating a range of cluster counts by the Elbow-Method using Within-Cluster-Sum-of-Squares, and present the results in Table 6.14.

The largest group, which we call the "High-sensitivity" group are the most privacy sensitive. Notably,

Participant group	# of Participants	Control	Awareness	Collection	Secondary Use	Avg Deny Rate
High-sensitivity	437	1.62	1.75	1.39	1.82	22.2%
Moderates	333	1.01	1.07	0.99	1.29	16.5%
Sharers	239	1.35	1.46	-0.06	1.59	11.2%
Low-sensitivity	167	0.68	0.53	0.01	0.18	12.4%

Table 6.14: Participant groups as clustered along 4 privacy dimensions

this group had the highest deny rate. The smallest group is the "Low-sensitivity" group, whose members were the least privacy sensitive and also had the low deny rate. In between the two, there is the "Moderates" group, whose scores across the 4 dimensions, as well as their deny rate, are between the High-sensitivity and the Low-sensitivity groups. However, we also found a fourth group, which we called the "Sharers" group, who are between the Moderates and the High-sensitivity in Control, Awareness and Secondary Use, but seem to be the most willing to share information, as indicated by them having the lowest sensitivity in the Collection dimension. This indicates some amount of independence between the 4 dimensions, and there is a significant group of participants in the Sharers group who are nearly as privacy-sensitive as those in the High-sensitivity group in all other dimensions other than Collection, where they are the least sensitive. It also indicates that privacy-sensitivity towards the collection of data is the largest indicator of an individual's deny rate relative to the general population, which makes sense since denying permissions generally reduces the amount of data that can be collected. We compared our groups to those in [28], and found that a rough mapping could be made, but because of the difference in methodology (their clusters are based on behavior while ours are based on attitudes), we are cautious in reading too much into the similarity. Nevertheless, the appearance of a rough mapping is worth noting.

Second, we have previously seen that participants are less likely to deny a permission request that they expected. We examine the distribution of correctly expected permission events in the 'paradoxical group' and plot that distribution versus the distribution of correctly expected permission events for the non-paradoxical participants (i.e. those with privacy sensitivity > 1.0 but deny rate $> 16.6\%$) in Figure 6.7. The histogram shows that while the paradoxical group grants more permissions, they also correctly expect those permissions more frequently than the non-paradoxical participants. While our data does not allows us to extract why this may be the case, we can speculate that the individuals in the paradoxical group may have a better understanding of the relationship between apps and the permissions requests, or perhaps they may be selecting applications that are more transparent about the permissions they require. In addition to the clustering above, this analysis helps explain why some of our participants exhibit an apparent paradox between their privacy attitude and permission denying behavior.

6.2 Complex Behaviors

We describe interesting and complex behaviors we observed, some of which we believe warrant future investigation.

6.2.1 Unexpected Yet Granted Requests

While participants were more likely to deny unexpected permission requests, over 70% of such permission requests were still granted. We observed that 32% of our participant pool (577 participants) had at least

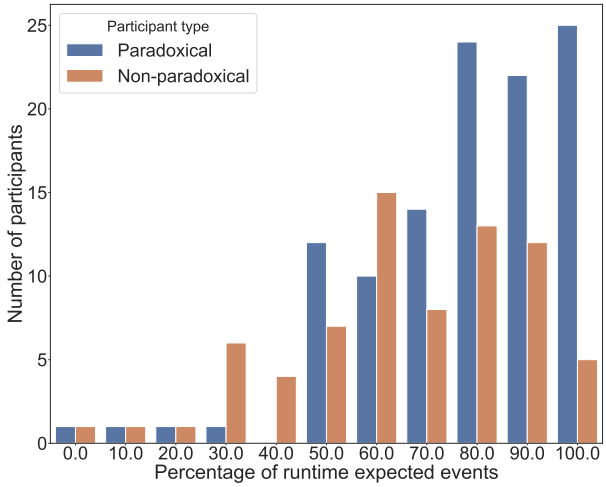

Figure 6.7: Histogram of the correct expectation rate of the paradoxical group vs. non-paradoxical participants

one unexpected permission request that they granted. To try to understand this behavior, we first looked at the reasons why these permission requests were granted. The top 3 reasons given were "I think the app won't work otherwise", (selected in 23% of such events), "I want to use a feature that needs this permission" (21%) and "I want the permission screen to go away" (18%). The first two reasons indicate that even though the participant didn't anticipate the request, the purpose appears to have become clearer during app usage. However, the third reason implies an impatient user who perhaps doesn't care much about privacy. We examined the users who selected this reason and found that 42% of this group is composed of members of the Sharers and Low-sensitivity groups. These two groups denied fewer permissions and were less concerned about the collection of private information. While this is a significant portion of the group, it is not the majority, and we believe that other influencing factors may be related to how apps interact with users, which may lead users to grant permissions even though they did not expect them. PrivaDroid did not collect detailed data on a participant's interaction with apps and we believe a separate study designed to collect such information is required to shed light on this complex and interesting behavior.

6.2.2 Expected Yet Denied Requests

Similarly, 385 (21%) participants denied a permission request they expected at runtime at least once. We sought to better understand this behavior by first looking at the deny reasons they gave when this occurred. "I can always grant it afterwards if I change my mind", leading the most popular reason at 33% of such events, followed by "I do not use the specific feature associated with the permission" at 28% and "I consider the permission to be very sensitive" at 18%. This data explains participants' thinking when denying an expected request from three angles: the capability to decide later, the usage of the app and the concern about sensitive data.

Comparing the deny reasons to when participants did not expect the request, we found that "I think the app shouldn't need this permission" became the top deny reason when participants did not expect it at 32%, double that of when they expected it. The fraction of "I do not use the specific feature associated with the permission" fell from 28% to 20%. Finally, the third top reason was "The app gave a poor

explanation", which was chosen in 13% of the deny surveys, compared to only 6% when participants expected the request. This again demonstrates the importance of explaining the reasons for a permission request to the user, either before the app is installed or at runtime during app use. Similar to grants of unexpected permission requests, we feel that the underlying reasons for this behavior depend on specific user-app interactions, which will be useful to study further.

Chapter 7

Covid-19 Impact

This chapter discusses the data of the participants from the 5 English speaking countries (i.e. US, Canada, UK, India and South Africa) and makes comparisons between participants in the *Pre-Covid* and *Post-Covid* group. We believe that Covid-19 may have an impact on user behaviors and their attitudes towards privacy and mobile apps because 1) users who are from the countries that enforce lockdown are spending more time at home, which may increase the chance of their home addresses being exposed by using apps that require *Location* permission, and 2) users may be encouraged to work from home, which requires installing more teleconferencing and productivity apps and granting the permissions associated with these apps. *Pre-Covid* means that participants completed the study before Covid-19 had not become a global pandemic whereas *Post-Covid* means they finished after Covid-19 had been recognized as a global pandemic. We use February 1st, 2020 as the date to determine whether a participant belongs to the *Pre-Covid* or the *Post-Covid* group. 950 participants from the *Pre-Covid* group finished our study and we refer to this group of participants as *Pre-Covid* group. These are 649 participants in the *Post-Covid* group, who successfully finished the study and we refer to these participants as *Post-Covid* group. Table 7.1 shows the detailed country and gender breakdown of participants from *Pre-Covid* and *Post-Covid* groups. Although we have a smaller group of participants than before, we reached at least 50 male and female participants in each country. Therefore, we believe our data is sufficient for drawing comparisons. We discuss some shifts of user behaviors around app install and permissions in the following sections.

7.1 App Install

Fewer applications installed. From the collected data, we observed that *Post-Covid* participants averaged fewer application installs than the *Pre-Covid* group with 33.9 and 36.1 respectively. One would assume that, with the Covid-19 lockdown policies and thus more time spent at home, people would have more time on their phones and try more new apps. However, that is not supported by our data. One possible explanation is that people mainly use their phones on their daily commute and now that they are home more, they have access to more entertainment resources such as laptops and televisions. Second reason for fewer installs could be that fewer new apps are put on market due to slowed down app development cycle. Although tech companies and software industries are reported to be less impacted by Covid-19, it is natural to think it takes some time for companies and engineers to

Country	Males		Females		Other		Did not say	
	Pre	*Post*	*Pre*	*Post*	*Pre*	*Post*	*Pre*	*Post*
United States	98	68	131	68	3	2	2	2
Canada	108	57	75	54	5	3	1	0
United Kingdom	85	71	55	51	0	1	0	1
India	202	102	60	56	0	0	0	1
South Africa	55	58	70	54	0	0	0	0
Total	548	356	391	283	8	6	3	4

Table 7.1: Country and Gender Demographics for *Pre-Covid* and *Post-Covid* Participants

adapt to this new work style. Another reason could be that people are spending more time on the old apps they installed instead of trying more new apps. The last reason may be because users become more privacy concerned about data collection practices by apps and choose to use fewer apps.

Even though the total number of app installs decreased slightly, we observe a surge of installs of telecommunication apps. We compared the fraction of participants that installed such apps from both *Pre-Covid* and *Post-Covid* groups. The apps that witnessed largest increases are Zoom (from 3.3% to 10.9%), Google Meet (from 0.8% to 4.7%), and Microsoft Teams (from 0.7% to 2.7%). This indicates that people seek more online meetings and conferencing instead of in-person meetings due to the social distancing policy. We also checked other apps in online shopping, productivity, food delivery and Covid-tracking categories. However, these was no notable uptake of installs in those categories.

Install reasons unchanged. We looked at the reasons why participants of the *Post-Covid* group installed apps and the top three most popular reasons remain the same as those of the *Pre-Covid* group. The top reasons are "I want to try it out" (37.9% of the surveys), "The app is useful" (24.1% of the surveys), and "The app is cool or fun to use" (17.9% of the surveys).

7.2 Permission Denial

The overall permission deny rate for the *Pre-Covid* group is 14.0%, which increased to 18.2% for the *Post-Covid* group. We explore how the deny rate changes from several perspectives.

7.2.1 Permission Categories

In order to answer why the deny rate increased by 4.2%, we first compared the deny rates of individual permission categories for these two groups of participants. Figure 7.1 shows the deny rates of each permission category (except *Body Sensors* and *Physical Activities* due to small number of permission events). All permission categories saw an increased or an unchanged deny rate except *Microphone*, which decreased from 30.9% to 28.7%. Deny rates for *Calendar* and *SMS* remain at 21.5% and 11.0% respectively. *Location* experienced the largest increase in deny rate, which is 9% from 11.2% to 20.2%, followed by *Camera* from 15.0% to 20.9%, *Contacts* from 17.5% to 22.8% and *Call Logs* from 7.8% to 12.9%. *Storage* and *Phone* both witnessed an increase approximately 2%. While *Microphone* stays the most denied permission, *Contacts* and *Location* moved to the second and fifth most denied permissions. The frequencies of each permission being requested are similar between *Pre-Covid* and *Post-Covid* group with a fluctuation of less than 3%.

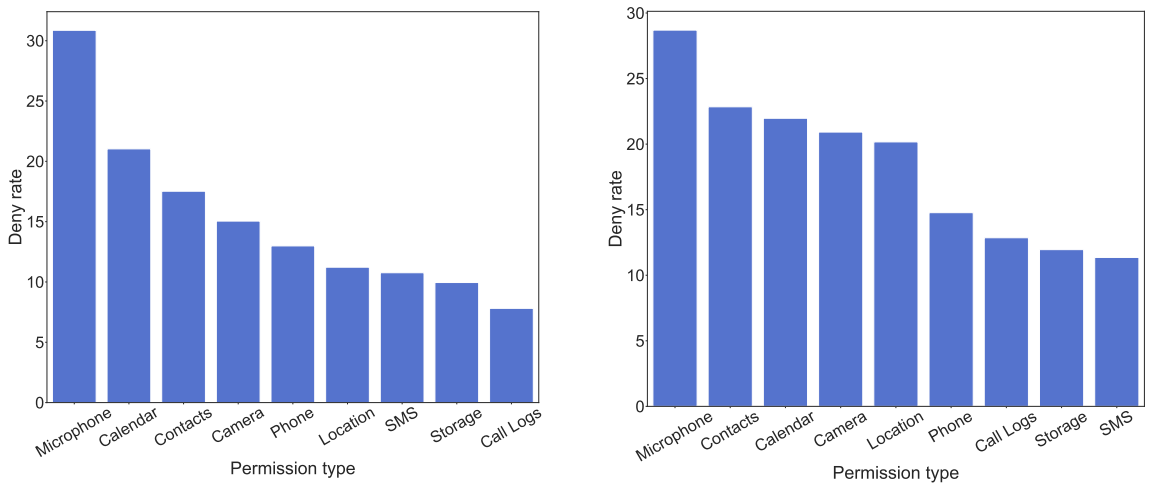

Figure 7.1: Deny rates of individual permission categories for *Pre-Covid* and *Post-Covid* groups

7.2.2 Demographics Analysis

We now look at the *Pre-Covid* and *Post-Covid* deny rates of individual permissions in the 5 countries. Figure 7.2 demonstrates the *Pre-Covid* and *Post-Covid* deny rates of individual permissions. We can see that deny rate for *Location* increased in all countries except South Africa with the largest jump in United States of 13%. *Contacts* permission was denied more frequently in United Status, United Kingdom and India after Covid-19 breakout.

We then move on to how deny rates varied for different genders. Before Covid-19, male participants denied 13.5% of the permission requests while female participants denied 14.5%. For *Post-Covid* group, both genders saw an increase in deny rates but in different amount. Deny rate for male participants increased to 15.3%, however, it climbed to 22.4% for female participants. We further look at deny rates of individual permission categories for both male and female participants to see where the biggest change happened. Deny rate for *Location* increased from 11.9% to 18.4% for male participants, followed by *Camera* and *SMS*, which witnessed an increase from 14.8% to 17.5% and 7.4% to 10.9% respectively. Deny rates for other permission categories have minimal changes. Female participants, on the other hand, denied more frequently all permissions except *Microphone* and *SMS*, whose deny rates dropped from 33.3% to 30.0% and from 17.8% to 14.3%. Permissions that experienced that largest increase for female participants happen to be the same as the three permissions with the largest increased deny rates overall, namely *Location*, *Contacts* and *Camera*. Deny rates of *Contacts* and *Location* doubled from 15.9% to 29.4% and from 10.1% to 21.1% for females. *Camera* deny rate also increased by almost 10% to 23.6%. It is obvious that female participants contributed more to the increased deny rates of *Location*, *Contacts* and *Camera*.

7.2.3 Privacy Attitude

Post-Covid participants are also required to complete an exit survey at the end of the study. We compare the privacy scores of these two groups in Table 7.3. Although we see that participants in the *Mid-Group* demonstrated a higher privacy sensitivity in all four categories, the difference is minimal with the largest increase of 0.07 in both Control and Secondary Use categories. We discussed in Section 6.1.3 that

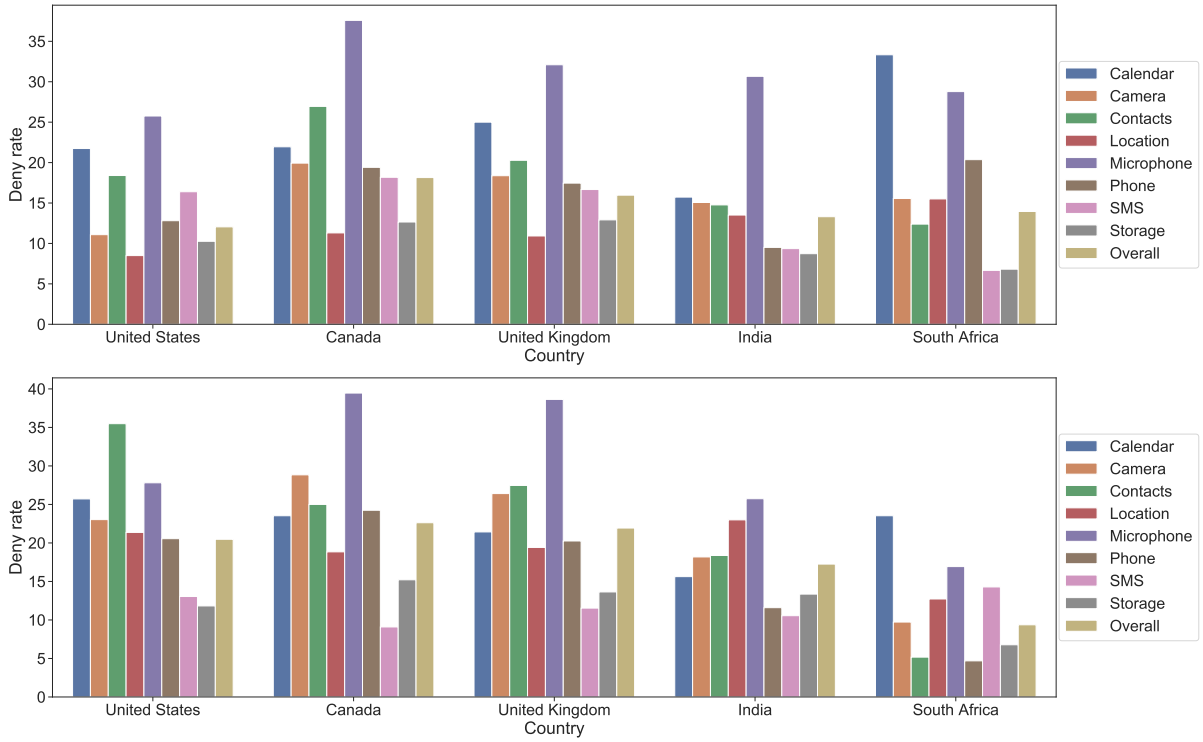

Figure 7.2: *Pre-Covid* (top) and *Post-Covid* (bottom) deny rates of permission types of each country

the overall privacy sensitivity increases, the average deny rate increases as well and thus we expect a slightly higher deny rate from the *Post-Covid* group. However, the deny rate increased by 4.2% while privacy sensitivity stayed relatively unchanged. We confirmed that participants from the two groups were similar in terms of demographics (e.g. gender, age, education level and income). Therefore, explanations other than privacy sensitivity and demographic influences are needed for the increased deny rate. One explanation could be that people become more conservative in face of the increased restrictions of their daily activities and heightened stress levels [9] caused by Covid-19.

Permission Type	Deny Rate of *Pre-Covid* Male Participants	Deny Rate of *Pre-Covid* Female Participants	Deny Rate of *Post-Covid* Male Participants	Deny Rate of *Post-Covid* Female Participants
Calendar	20.8%	22.7%	20.3%	19.4%
Camera	14.8%	14.4%	17.5%	23.6%
Contacts	18.1%	15.9%	18.4%	29.4%
Location	11.9%	10.1%	18.4%	21.1%
Microphone	29.9%	33.3%	27.5%	30.0%
Phone	11.9%	14.9%	10.9%	21.3%
SMS	7.4%	17.8%	10.9%	14.3%
Storage	8.4%	12.7%	8.8%	15.9%
Overall	13.5%	15.3%	14.5%	22.4%

Table 7.2: Permission deny rates of males and females in *Pre-Covid* and *Post-Covid* groups

Group type	Control	Awareness	Collection	Secondary Use	Overall
Pre-Covid	1.29	1.36	0.76	1.41	1.19
Post-Covid	1.36	1.41	0.82	1.48	1.25

Table 7.3: Privacy scores of *Pre-Covid* and *Post-Covid* groups

Chapter 8

Limitations

Due to the nature of our participant recruitment, which relies on online advertising, our study is biased towards users who interact with online ads. We are unable to collect data from potential participants who do not interact with online ads or who may be unwilling to install PrivaDroid. In addition, because our study requires participants to install PrivaDroid and grant it accessibility and app usage permissions, we expect there to be a selection bias in our pool of participants. To measure this bias, we compare the average responses to the privacy sensitivity questions in the exit survey with those from the 100 Amazon Mturk workers we used to measure the quality of our questions using Chronbach's Alpha, and who did not install PrivaDroid (though they were willing to use the MTurk platform). We tabulate the comparison in Table 8.1. The results confirm that there is some selection bias in the participants, as the MTurk worker's scores are higher across all 4 dimensions (meaning they are more privacy sensitive) and significantly higher in the Collection dimension. During our study, we found that males were more likely to join our study and to achieve a balanced gender split, we targeted our advertising towards females on Facebook and Reddit first. For Reddit, we targeted certain interests, which may result in more females that interact with content around those interests joining our study.

PrivaDroid cannot collect data on events that occurred before it was installed, thus we do not see any permission decisions participants made with their apps before the start of our study. It may thus under-count events caused by the default apps that come with a phone, or popular applications that are likely to have been already installed on a participant's phone. Users may have different behaviors around the pre-installed applications and this can be addressed by bringing people into the lab and studying their behaviors. Both participant bias and blindness to pre-install events are unavoidable side-effects of our recruitment and data collection methodologies. Although 1,780 is a large number of participants for a user study, this is of course, also a small sample across the world population. All findings reported here clearly reflect the behavior of our participants and do not reflect the populations of our study countries at large. Also, even if we aimed for a balanced split of male and female participants, we could not control and know for certain that if the participants that already joined would stay for the entire

User type	Control	Awareness	Collection	Secondary Use
MTurker	1.62	1.35	1.53	1.59
Participants	1.26	1.32	0.78	1.39

Table 8.1: Privacy scores of MTurk workers and our participants

30 days. Therefore, we decided to recruit more participants than we needed to have some buffer room when we advertised our app. This resulted into a larger participant pool with imbalanced gender split.

In addition, 42% of the users participating in our study did so after March 15, 2020, when the social and economic measures caused by the Covid-19 pandemic came into force in the majority of the countries in our study, and we are unable to conclusively ascertain the effect of those measures on this group of participants.

Lastly, participants communicated to us that they received warnings on their devices saying that PrivaDroid consumes too much battery. This may explain why some participants dropped out during the experiment. We pinpointed the cause to be a combination of the use of *foreground service* that is always running to monitor app install and app removal events and accessibility service that constantly checking whether participants encountered a runtime permission request or changed permission settings in the Settings menu. However, we did not find a solution to circumvent these issues due to the Android system restrictions and our methodology.

Chapter 9

Future Work

Although we made some very interesting discoveries in behaviors of our participants around Android permission and developed more insights in their attitudes towards privacy, there are still many dimensions of the data that yet to be explored due to the lack of time. For example, we collected the demographic information about participants' daily phone usage time and their employment status, but we did not perform any data analysis based on those categories. Another example is that we did not check what type of apps are participants in different demographic group more likely to install and which permissions they tend to deny.

Although we designed our survey questions carefully to answer the research questions we had prior to the experiment, the data we collected often generates new research questions and interests, some of which the current data is not enough to answer. For example, we discussed some complex behaviors of our participants in Section 6.1.3 but we were not able to explain them without additional data. These complex behaviors require but not limited to the context in which participants were using the app. We tried to understand the context better by capturing the rationale messages apps provide before permission requests. We used a keyword based heuristics to find potential rationale messages but our heuristics can be inaccurate and it is not enough to capture all the rationale messages. For example, if apps provide rationales in an image or the app flow is well designed so there is no need for a rationale, PrivaDroid will not be able to capture those.

Another potential application of the data we collected is to model the user's privacy preference and make recommendations to the user. Felt et al. [19] and Lin et al. [28] conducted surveys and interviews in order to better understand mobile users' concerns around mobile privacy and model people's mobile app privacy preferences. We have the data about which permissions users denied for what type of apps, which permissions users had a decent understanding on and accurate expectation of, users' privacy attitudes towards mobile privacy and etc. so that we can model their privacy preferences.

Lastly, we recruited participants from the 5 English speaking countries in order to compare user behaviors and privacy attitude change caused by Covid-19. The participants finished the experiment in late July so it left us little time to dig into the data. We performed some simple analysis on the data but we expect to analyze the data more and publish our findings after more detailed analysis.

Chapter 10

Conclusions

We found that a few trends reported in [16] remain the same three years later: the aggregate denial rate still hovers around 16-17%, *Microphone* is still the most often denied permission, and we continue to see variation in deny rates across the permission types. At the same time, there were some notable changes for specific permission types. For example, the deny rate for the *Calendar* permission has grown significantly from 10% [16] to 24.2% today and the deny rates for the *Phone* permission have dropped significantly from 19% to 12.8%. We do not know why this change of behavior with the *Calendar* permission has occurred.

Our demographic analysis reveals interesting trends across countries. We observed that the intra-country variance of deny rates is much higher than the inter-country variance. This suggests that rather than countries having fundamentally different views towards privacy, countries with large populations are more likely to have residents whose behaviors with privacy controls vary broadly across the spectrum. At the same time, Argentina and Spain have aggregate deny rates that are twice that of countries such as the US and South Africa. In terms of specific permissions, our 5 English language countries all show *Microphone* as the permission that is clearly denied most. Yet France and Singapore are far more sensitive to the *Calendar* permission than the others. Argentina seems roughly equally sensitive to all permission types. The 2017 study reported that, in the US, women deny permissions nearly twice as often as men. This trend does not hold in this global study, where we found that women deny permissions (18%) only a little more than men (16%).

Our study revealed that including explanations for permissions brings strong benefits to apps as it reduces the deny rate by more than half (17% for apps without explanations and 7% for those providing explanations). We also recorded two types of user expectations of permission requests: expectations at app install time and expectations at runtime right after a permission request. In both cases, we found that participants deny permissions more often when an app asks for a permission they did not expect. This bias exists for both types of expectations and across all permission types, but is significantly stronger for runtime expectations, where the deny rate for unexpected permissions is double that of expected permissions. We also learned that users are generally quite poor at predicting if an app will ask for a particular permission; we found they guessed incorrectly over two thirds of the time.

Our study enabled us to assess whether participants "do what they say" since we captured both privacy attitudes via a survey and actual behaviors on participants personal devices. We find overall that as self-stated privacy sensitivity increases, the average permission deny rates increases as expected, but

interestingly, so does the variance in deny rates. We observed that roughly a quarter of our participants exhibited the privacy paradox behavior in that they claimed they were privacy sensitive but had low deny rates. We found that expectations can explain some of this seemingly paradoxical behavior. These participants had a much greater ability to correctly predict (expect) which permissions an app might request. (This was in contrast to the majority of users whose expectations were poorly aligned with what apps request.) This suggests that users who are privacy sensitive yet have low deny rates may be limiting their app choices and smartphone uses to scenarios they understand well. We leave further exploration of this app selection hypothesis to future work.

Appendix A

Appendix

A.1 Consent Form

Here we attach the English version of the consent form, which is translated into Spanish, French and Traditional Chinese.

English consent form. This experiment is entirely voluntary and you can join/exit the experiment at any given time. Participants must be over 18 years old. The purpose of this experiment is to study the reasons for the decisions users make when answering runtime permission requests and when installing and uninstalling applications. Survey questions will appear at some app install and uninstall events, and sometimes at the moment of a permission request. We collect demographic information once at the beginning and ask for your opinion on a few items in an exit survey. The app collects no personally identifiable data except for your Google Advertising ID and we will not use this ID to re-identify you. Your data and answers are only shared with our research team and access control is used to secure your data.

Participants need to enable accessibility service and app usage access for PrivaDroid to work properly. Participants who answer the demographic questionnaire and stay in the experiment for 30 days will qualify for a 10 USD reward in the form of PayPal payment. We ONLY support PayPal payment method. Multiple enrollments in this experiment only qualify for one payment. Payments may take about three weeks and longer. Currently, we only accept participants from Canada, United States, United Kingdom, South Korea, Spain, France, Hong Kong, India, South Africa, Argentina and Singapore. Users outside of the above countries can still participate but will not be eligible to receive a reward. During the experiment, your Android language MUST be set to one of the following languages: English, Spanish, French, Korean and Traditional Chinese. You won't be eligible for the reward otherwise. The University of Toronto received donations from Google for this project. However, no raw data collected from this research will be shared with Google or any of its staff. Thanks for joining!

A.2 Survey Questions

Here we list the English version of the survey questions and available options.

A.2.1 Demographic Survey

Users were required to answer all questions but were allowed to select the "Prefer not to say" option.

What is your age?

- Below 20
- Between 20 and 30
- Between 30 and 40
- Between 40 and 50
- Between 50 and 60
- Above 60
- Prefer not to say

What is your gender?

- Male
- Female
- Other
- Prefer not to say

Which country do you live in?

List of all countries.

What is your income level in USD?

- $0
- $1 - $9999
- $10000 - $24999
- $25000 - $49999
- $50000 - $74999
- $75000 - $99999
- $100000 - $149999
- $150000 and above
- Prefer not to say

What is the highest degree or level of school you have completed?

- Less than a high school diploma
- High school degree or equivalent
- Bachelor's degree (e.g. BA, BS)

- Master's degree (e.g. MA, MS)
- Doctorate (e.g. PhD)
- Prefer not to say

Which of the following categories best describes the industry you primarily work in?

- Agriculture, Forestry, Fishing and Hunting
- Arts, Entertainment, and Recreation
- College, University, and Adult Education
- Computer and Electronics Manufacturing
- Construction
- Finance and Insurance
- Government and Public Administration
- Health Care and Social Assistance
- Hotel and Food Services
- Legal Services
- Manufacturing
- Others
- Real Estate, Rental and Leasing
- Retail
- Scientific or Technical Services
- Software
- Transportation and Warehousing
- Unemployed
- Prefer not to say

What is your current employment status?

- Employed full time (40 or more hours per week)
- Employed part time (up to 39 hours per week)
- Unemployed and currently looking for work
- Unemployed and not currently looking for work
- Student
- Retired
- Self-employed
- Unable to work
- Other
- Prefer not to say

How much time do you spend on your phone daily?

- Less than an hour
- Between 1 and 2 hours
- Between 2 and 3 hours
- Between 3 and 4 hours
- Between 4 and 5 hours
- Between 5 and 6 hours
- Over 6 hours
- Prefer not to say

A.2.2 App Install Event Survey

We ask users right after they install an app about why they installed the app and they can select from a list of provided reasons. We ask users right after they install an app about which permissions they expect the app will ask for. Participants can choose as many as they like.

Why did you install the app?

- I want to try it out
- The app is useful
- The app is part of a product/service I use
- The app is cool or fun to use
- I trust the app or the company making the app
- My friends/family use it
- It was the only app of its kind (no other apps provide the same functionality)
- I was required to install it
- I was offered something in return (e.g. credits, monetary rewards, discount)
- The app has fewer permissions than other apps like it
- None
- Other

Which factors influenced your decision to install the app?

- App rating
- App popularity
- Individual user reviews
- Requested permission
- The company creating the app
- The app is free / price is reasonable

- App functionality
- None
- Other

Do you know what permissions this app requires?

- Yes
- No
- Not sure

Which of the following permission do you think the app requires?

- Camera
- Contacts
- Location
- Microphone
- Phone
- Storage
- Body Sensors
- Calendar
- SMS
- Call Logs
- Physical Activity
- None
- I don't know

A.2.3 App Removal Event Survey

After participants remove an app, we ask them why they remove the app and what permissions they remember the app asked for. We randomized the order of the possible options except for the "None" and "Other", which were always placed at the end.

Why did you uninstall the app?

- I no longer use the app
- To free up space or speed up my device
- I didn't like the app
- The app is not working as expected
- The app is crashing/very slow
- Because of advertisements in the app

- Because of in-app purchases
- The app required permissions I wasn't comfortable granting
- None
- Other

Do you remember if the app requires any of the following permissions?

- Camera
- Contacts
- Location
- Microphone
- Phone
- Storage
- Body Sensors
- Calendar
- SMS
- Call Logs
- Physical Activity
- None
- I don't remember

A.2.4 Permission Grant Event Survey

We randomized the order of the possible options except for the "None" and "Other", which were always placed at the end.

Why did you grant the permission request?

- I want to use a feature that needs this permission
- I trust the developer
- I think the app won't work otherwise
- I have nothing to hide
- The developer already has this information about me
- I want the permission screen to go away
- Because the app is popular
- The app gave an explanation that made sense
- None
- Other

The following question is used to gauge permission expectations at runtime.

Did you expect the app requests this permission?

- Yes
- No

How comfortable do you feel granting this permission request?

- Very uncomfortable
- Somewhat uncomfortable
- Neutral
- Somewhat comfortable
- Very comfortable

Do you want to grant the permission temporarily?

- Yes
- No

Would you like a notification to remind you of disabling this permission later?

- No
- In 1 hour
- In 2 hours
- In 4 hours
- In 8 hours
- In 24 hours

A.2.5 Permission Denial Event Survey

We randomized the order of the possible options except for the "None" and "Other", which were always placed at the end.

Why did you deny the permission request?

- I think the app shouldn't need this permission
- I can always grant it afterwards if I change my mind
- I do not use the specific feature associated with the permission
- I consider the permission to be very sensitive
- I don't trust the developer
- I wanted the permission screen to go away
- The app gave a poor explanation
- I think something bad might happen

- None
- Other

Did you expect the app requests this permission?

- Yes
- No

A.2.6 Exit Survey

Participants were asked to state how much the agree or disagree with each statement in this survey using the following 5 options:

- Strongly Agree
- Agree
- Neither Agree Nor Disagree
- Disagree
- Strongly Disagree

Note that question 4 in the Control section (A.5.1) is in direct opposition to the statement in question 4 of the Collection Section ((A.5.3). This was inserted as an attention checking question. Surveys with contradictory answers were not used.

Control Section Questions

1. Mobile app privacy is about a user's right to exercise control over decisions about how their information is collected, used, and shared.
2. User control of personal information is essential to mobile app privacy.
3. I believe that mobile app privacy is compromised when the user loses control over their information as a result of app usage.
4. I'm not concerned that smartphone apps are collecting too much personal information about me[1].

Awareness of Privacy Practices Section Questions

1. Mobile app developers seeking information should disclose the way the data are collected, processed, and used.
2. A good mobile app privacy policy should have a clear and conspicuous disclosure.
3. It is very important to me that I am aware and knowledgeable about how my personal information will be used.

[1]This is the attention-checking question.

Collection Section Questions

1. It usually bothers me when smartphone apps ask me for personal information.
2. When mobile apps ask me for personal information, I sometimes think twice before providing it.
3. It bothers me to give personal information to so many mobile apps.
4. I'm concerned that smartphone apps are collecting too much personal information about me.

Secondary Use Section Questions

1. Mobile apps should not use personal information for any purpose unless it has been authorized by the individuals who provided information.
2. When people give personal information to a mobile app for some reason, the app developer should never use the information for any other reason.
3. Mobile app developers should never sell the personal information in their computer databases to other companies.
4. Mobile app developers should never share personal information with other companies unless it has been authorized by the individual who provided the information.

Additional Questions

1. How familiar are you with the Android permission system?
    - Very Familiar
    - Somewhat Familiar
    - Not Very Familiar
    - Never Heard of This
2. Are there android permissions you do not understand what it means or what it does? Please select all permissions that you do not understand.
    - Camera
    - Contacts
    - Location
    - Microphone
    - Phone
    - Storage
    - Body Sensors
    - Calendar
    - SMS
    - Call Logs
    - Physical Activity
    - None

Bibliography

[1] Best practices for unique identifiers | Android Developers. https://developer.android.com/training/articles/user-data-ids, 2019. Accessed: July 2020.

[2] Request App Permissions — Android Developers. https://developer.android.com/training/permissions/requesting, 2019. Accessed: July 2020.

[3] Advertising ID | Play Console Help. https://support.google.com/googleplay/android-developer/answer/6048248?hl=en, 2020. Accessed: July 2020.

[4] Android Developers. https://developer.android.com, 2020. Accessed: July 2020.

[5] Android framework classes and services. https://android.googlesource.com/platform/frameworks/base/, 2020. Accessed: August 2020.

[6] Coronavirus disease (COVID-19) pandemic. https://www.who.int/emergencies/diseases/novel-coronavirus-2019, 2020. Accessed: July 2020.

[7] Firebase. https://firebase.google.com/, 2020. Accessed: June 2020.

[8] A Guide to In-App Advertising. https://www.smaato.com/guide-to-in-app-advertising/, 2020. Accessed: July 2020.

[9] Mental Health and Coping During COVID-19. https://www.cdc.gov/coronavirus/2019-ncov/daily-life-coping/managing-stress-anxiety.html, 2020. Accessed: August 2020.

[10] Mobile Advertising. https://theonlineadvertisingguide.com/ad-size-guide/mobile-advertising/, 2020. Accessed: July 2020.

[11] PackageInstaller. https://android.googlesource.com/platform/packages/apps/PackageInstaller, 2020. Accessed: August 2020.

[12] Settings. https://android.googlesource.com/platform/packages/apps/Settings/, 2020. Accessed: August 2020.

[13] What does Cronbach's Alpha mean? https://stats.idre.ucla.edu/spss/faq/what-does-cronbachs-alpha-mean/, 2020. Accessed: August 2020.

[14] Alessandro Acquisti and Jens Grossklags. What can behavioral economics teach us about privacy. *Digital privacy: theory, technologies and practices*, 18:363–377, 2007.

[15] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor, and Yuvraj Agarwal. Your location has been shared 5,398 times!: A field study on mobile app privacy nudging. In *Proceedings of CHI*. ACM, 2015.

[16] Bram Bonné, Sai Teja Peddinti, Igor Bilogrevic, and Nina Taft. Exploring decision making with android's runtime permission dialogs using in-context surveys. In *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*, pages 195–210, Santa Clara, CA, July 2017. USENIX Association.

[17] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. Android permissions demystified. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS. ACM, 2011.

[18] Adrienne Porter Felt, Serge Egelman, Matthew Finifter, Devdatta Akhawe, and David Wagner. How to ask for permission. In *Proceedings of 7th Usenix conference on Hot Topics in Security (HotSec)*, 2012.

[19] Adrienne Porter Felt, Serge Egelman, and David Wagner. I've got 99 problems, but vibration ain't one: A survey of smartphone users' concerns. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM. ACM, 2012.

[20] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, SOUPS. ACM, 2012.

[21] Marian Harbach, Markus Hettig, Susanne Weber, and Matthew Smith. Using personal examples to improve risk communication for security & privacy decisions. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI. ACM, 2014.

[22] Corey Brian Jackson and Yang Wang. Addressing the privacy paradox through personalized privacy notifications. *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, 2(2):1–25, 2018.

[23] Jaeyeon Jung, Seungyeop Han, and David Wetherall. Short paper: Enhancing mobile application permissions with runtime feedback and constraints. In *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM. ACM, 2012.

[24] Patrick Gage Kelley, Sunny Consolvo, Lorrie Faith Cranor, Jaeyeon Jung, Norman Sadeh, and David Wetherall. A conundrum of permissions: Installing applications on an android smartphone. In *Proceedings of the 16th International Conference on Financial Cryptography and Data Security*, FC. Springer-Verlag, 2012.

[25] Patrick Gage Kelley, Lorrie Faith Cranor, and Norman Sadeh. Privacy as part of the app decision-making process. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI. ACM, 2013.

[26] Spyros Kokolakis. Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon. *Computers & security*, 64:122–134, 2017.

[27] Jialiu Lin, Shahriyar Amini, Jason I. Hong, Norman Sadeh, Janne Lindqvist, and Joy Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, UbiComp. ACM, 2012.

[28] Jialiu Lin, Bin Liu, Norman Sadeh, and Jason I. Hong. Modeling users' mobile app privacy preferences: Restoring usability in a sea of permission settings. In *10th Symposium On Usable Privacy and Security (SOUPS 2014)*, pages 199–212, Menlo Park, CA, July 2014. USENIX Association.

[29] Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun (Aerin) Zhang, Norman Sadeh, Yuvraj Agarwal, and Alessandro Acquisti. Follow my recommendations: A personalized privacy assistant for mobile app permissions. In *Twelfth Symposium on Usable Privacy and Security*, SOUPS. USENIX Association, 2016.

[30] X. Liu, Y. Leng, W. Yang, W. Wang, C. Zhai, and T. Xie. A large-scale empirical study on android runtime-permission rationale messages. In *2018 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 137–146, 2018.

[31] Xueqing Liu, Yue Leng, Wei Yang, Chengxiang Zhai, and Tao Xie. Mining Android app descriptions for permission requirements recommendation. In *IEEE International Requirements Engineering Conference*, pages 147–158, 08 2018.

[32] N. Malhotra, S. Kim, Rosson, and J. Agarwal. Internet Users' Information Privacy Concerns (IUIPC): The Construct, the Scale and a Causal Model. *Information Systems Research*, December 2004.

[33] Nathan Malkin, Julia Bernd, Martiza Johnson, and Serge Egelman. What can't data be used for? privacy expectations about smart tvs in the usa. In *European Workshop on Usable Security (EuroSEC)*, 2018.

[34] Andrew McNamara, Akash Verma, Jon Stallings, and Jessica Staddon. Predicting mobile app privacy preferences with psychographics. In *Proceedings of the 2016 ACM on Workshop on Privacy in the Electronic Society*, WPES '16, page 47–58, New York, NY, USA, 2016. Association for Computing Machinery.

[35] Helen Nissenbaum. Privacy as contextual integrity. *Washington Law Review*, 2004.

[36] Patricia A Norberg, Daniel R Horne, and David A Horne. The privacy paradox: Personal information disclosure intentions versus behaviors. *Journal of consumer affairs*, 41(1):100–126, 2007.

[37] René Proyer and Joachim Häusler. Gender differences in vocational interests and their stability across different assessment methods. *Swiss Journal of Psychology*, 66, 12 2007.

[38] Jonathan Schubauer, David Argast, and L Jean Camp. Lessig was right: Influences on android permissions. In *TPRC48: Research Conference on Communications, Information and Internet Policy*, 2018.

[39] Irina Shklovski, Scott D. Mainwaring, Halla Hrund Skúladóttir, and Höskuldur Borgthorsson. Leakiness and creepiness in app space: Perceptions of privacy and mobile app use. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems*, CHI. ACM, 2014.

[40] Joshua Tan, Khanh Nguyen, Michael Theodorides, Heidi Negrón-Arroyo, Christopher Thompson, Serge Egelman, and David Wagner. The effect of developer-specified explanations for permission requests on smartphone user behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, page 91–100, New York, NY, USA, 2014. Association for Computing Machinery.

[41] Permission updates in Android 11: One-time permissions. `https://developer.android.com/preview/privacy/permissions`, 2020. Accessed: June 2020.

[42] Christopher Thompson, Maritza Johnson, Serge Egelman, David Wagner, and Jennifer King. When it's better to ask forgiveness than get permission: Attribution mechanisms for smartphone resources. In *Proceedings of the Ninth Symposium on Usable Privacy and Security*, SOUPS. ACM, 2013.

[43] Janice Y Tsai, Serge Egelman, Lorrie Cranor, and Alessandro Acquisti. The effect of online privacy information on purchasing behavior: An experimental study. *Information systems research*, 22(2):254–268, 2011.

[44] Lynn Tsai, Primal Wijesekera, Joel Reardon, Irwin Reyes, Serge Egelman, David Wagner, Nathan Good, and Jung-Wei Chen. Turtle guard: Helping android users apply contextual privacy preferences. In *Symposium on Usable Privacy and Security (SOUPS)*, 2017.

[45] Timothy Vidas, Nicolas Christin, and Lorrie Cranor. Curbing android permission creep. In *Proceedings of the Web*, volume 2, pages 91–96, 2011.

[46] Ana Villar. Agreement answer scale design for multilingual surveys: Effects of translation-related changes in verbal labels on response styles and response distributions. 2009.

[47] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. Android permissions remystified: A field study on contextual integrity, 2015.

[48] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner, and Konstantin Beznosov. The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences. In *Proceedings of the 38th IEEE Symposium on Security and Privacy*. IEEE, 2017.