On a (Per)Mission: Leveraging User Ratings of App Permissions to Help Users Manage Privacy

by
Hannah Quay-de la Vallee
B. S., Bard College, 2010
Sc. M., Brown University, 2013

A dissertation submitted in partial fulfillment of the
requirements for the Degree of Doctor of Philosophy
in the Department of Computer Science at Brown University

Providence, Rhode Island
May 2017

This dissertation by Hannah Quay-de la Vallee is accepted in its present form by the Department of Computer Science as satisfying the dissertation requirement for the degree of Doctor of Philosophy.

Date _____ _____

Shriram Krishnamurthi, Director

Recommended to the Graduate Council

Date _____ _____

Jeff Huang, Reader

Date _____ _____

Michael Littman, Reader

Approved by the Graduate Council

Date _____ _____

Andrew G. Campbell
Dean of the Graduate School

Contents

List of Figures

Abstract of "On a (Per)Mission: Leveraging User Ratings of App Permissions to Help Users Manage Privacy" by Hannah Quay-de la Vallee, Ph.D., Brown University, May 2017.

Apps provide valuable utility and customizability to a range of user devices, but installation of third-party apps also presents significant security risks. Many app systems use permissions to mitigate this risk. It then falls to users to decide which apps to install and how to manage their permissions, but unfortunately, many users lack the expertise to do this in a meaningful way.

In this thesis, I determine that users face two distinct privacy decisions when using apps: which apps to install, and how to manage apps' permissions once they are installed. In both cases, users are not given meaningful guidance to help them make these choices.

For decisions about which apps to install, users would benefit from privacy information in the app marketplace, since that is how most users choose apps. Once users install an app, they are confronted with the second type of decision: how to manage the app's permissions. In this case, users would benefit from an assistant that helps them see which permissions might present privacy concerns. I therefore present two tools: a privacy-conscious app marketplace and a permission management assistant.

Both of these tools rely on privacy information, in the form of ratings of apps' permissions. I discuss gathering this rating information from both human and automated sources and how it is used in the two tools. I also explore how the brand of an app could affect how users rate its permissions. Additionally, because my goal is to convey privacy information to users, I design and evaluate several interfaces for displaying permission ratings. I discuss surprising misconceptions generated by some of these interfaces, and present an interface that effectively communicates permission ratings.

Chapter 1

Introduction

Thesis Statement *To encourage users to use more privacy-respecting apps, app stores should include user ratings of privacy as a criterion for sorting apps. In the absence of user-provided ratings, crowdsourcing can be used to gather ratings. Additionally, these ratings can be leveraged to help users determine which permissions to enable and disable for a given app.*

App-based devices have become pervasive in consumers' lives [2], due in part to apps' easy installation model. Most app ecosystems are supported by a central marketplace that enables users to easily search for, investigate, and install apps, allowing users of all levels of technical ability to customize their devices. However, the amount of user information associated with these devices makes third-party apps a threat to user security and privacy. Indeed, apps have become a popular target of hackers and malware developers, leading to exposure of private information and financial harm [20, 21, 30, 47]. In addition to malware, there are apps that, while not necessarily malicious, collect significant amounts of user data, which can be be sold to other companies, used to target ads, or otherwise used in ways users did not expect and may not approve of [15, 29, 40, 53, 54]. The expansion of the app model beyond smartphones to platforms such as desktops, cars, and the Internet of Things exacerbates these concerns [4, 28, 32, 34].

Many platforms try to mitigate these risks by requiring users to grant permission before apps can access certain hardware resources and user data. Unfortunately, such systems force users, even technical novices, to manage their own privacy without assistance. Furthermore, most systems ask for consent either at or after installation time, when users have already chosen an app, making it onerous for them to switch apps if they dislike an app's permission requests.

App stores, as a primary source of app information, are ideally positioned to act as a fulcrum to aid users in managing their privacy.[1] In fact, marketplaces *already* influence user decisions by ranking apps and thus filtering which apps users see. Unfortunately, marketplaces are not incentivized to put user privacy first. It is even possible, given the profit model of many app stores, that apps that use more ad libraries are put first.

Apart from baseline protection like malware detection, most app marketplaces do not use their position to better inform or protect users. Google Play, Android's proprietary marketplace, allows users to search by price and star rating but does not provide privacy-based search options, nor any privacy guidance past simply listing apps' permissions with brief descriptions (which are vague enough that they have been the subject of lawsuits by users arguing that they were not adequately informed of apps' capabilities [30]). As a result, many users can only give *uninformed consent* to permission requests, as they are ill-equipped to judge whether an app's permissions are appropriate for its purpose.

Worse, users who *do* have judgements about apps' permissions have no good way to express themselves to other users or to developers. Some try to communicate their opinions via app reviews, but these are difficult to find amongst the myriad reviews. Worse still, developers who *want* to explain their app's permission requests also lack a dedicated forum to do so. Some developers use their app's description page but, since this is not standard practice, it is easy for users to miss, especially in Google Play, where long descriptions are hidden by default. Other apps, like Pinterest, explain their permission requirements on their websites [7], where only a very motivated user is likely to find it.

Because it is not standard for developers to explain their permission requests, it is not generally seen as suspicious when they do not. This is concerning since some permissions grant access to large chunks of information, and the user does not know what information the app is collecting or how it is using that information. For instance, the Uber ride-sharing app uses the "read phone status and identity" permission to view the battery status to know when to switch to low power mode. However, it also uses that information for research, so Uber now knows that users with low battery are more willing to pay higher prices for rides. Although Uber says it does not adjust price based on battery status, it may come as a surprise to some users that they know that at all [12]!

Despite this quagmire, user reviews *have* been a useful privacy tool, harnessed by researchers to inform

[1] In fact, app stores play such a key role in how users discover apps that they have been the target of censorship in countries like China and Russia [38].

users about the consequences of updating their existing apps [49], and by developers to read user opinion and guide app development. For instance, an update of the Avis car rental app added the "retrieve running apps" permission. This led users to leave a spate of negative reviews, spurring the app's developers to remove the permission. These examples show that reviews can be a valuable source of information about app permissions, but current marketplaces limit their effectiveness by making them difficult for users and developers to find.

We have built two Android apps that leverage privacy rating information to help users make informed privacy decisions. We also allow developers to respond to these ratings, thereby providing a channel for communication between users and developers. The first app is a privacy-conscious marketplace, which helps users to find privacy-respecting apps. The second is a permission management assistant to help users regulate their apps' permissions after they are installed.

Applicability Beyond Android We focused on Android because it offered a concrete platform with a broad user base on which we could build real-world functional tools, and because there is a significant body of academic research on the Android permission system. However, the *concepts* of this work could apply to any app platform, such as Chrome browser extensions, car app stores [26], and Internet of Things apps [1], and, of course, other mobile platforms like iOS and Windows Phone. Different platforms may need to adjust specifics like the exact interface for presenting ratings, or how privacy influences ranking, but the broader concept of privacy information in the marketplace is applicable across the spectrum of app platforms.

Contributions This thesis makes several contributions. In Chapter 3, we show that users face privacy decisions both when selecting which apps to install and when managing their apps after installation. Second, in Chapter 5 we use crowdsourcing and automated tools to collect ratings of apps' permissions to assist users with their privacy decisions. In Chapter 6 we examine how brand may affect user ratings, and in Chapter 7 we show how these ratings can be used to promote privacy-respecting apps in a marketplace. In Chapter 8 we discuss the crowd feedback-based method we used to develop an interface for presenting rating information to users, including some unexpected subtleties in the design of such an interface. All of these features are incorporated into our two apps, which are discussed in Chapter 4. Chapter 2 provides an overview of different permission models, and Chapter 9 discusses related work.

Chapter 2

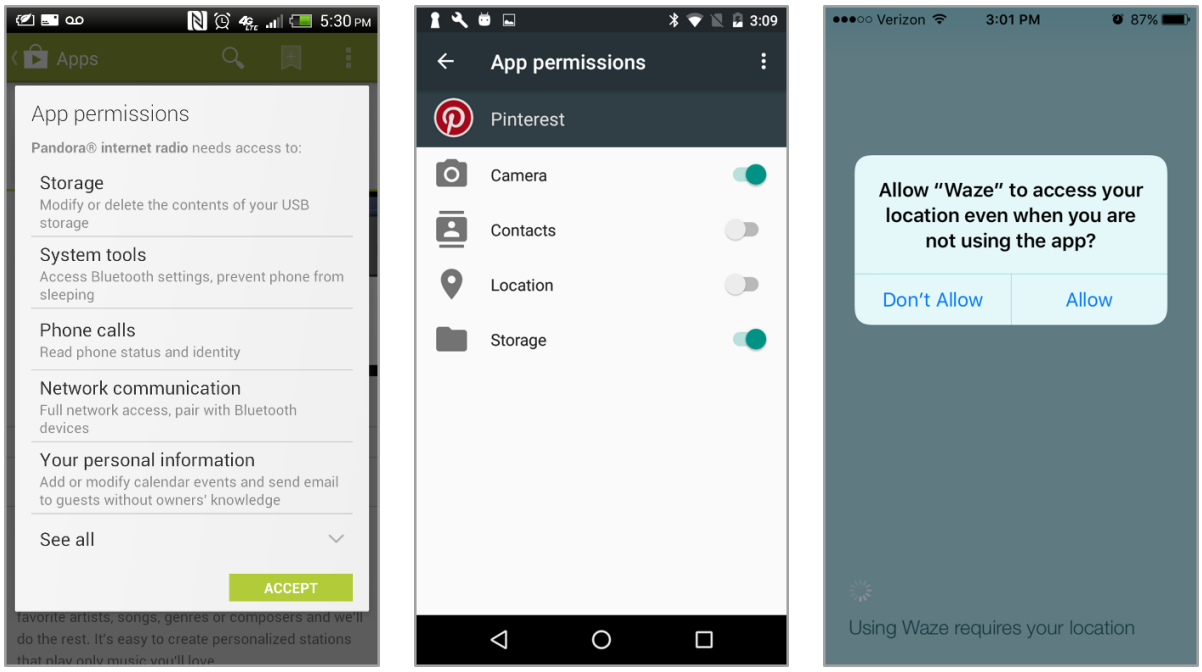
Overview of Permission Models

While permissions are a common method of restricting apps' access to resources and data, there are significant variations between platforms in how those permissions work. For instance, some platforms, such as iOS, allow users to decide *when* an app may use certain permissions, not just *if*, while in other platforms offer only a binary decision of grant or deny.

Perhaps the most significant variation is an "all-or-nothing" model, where an app requires some set of permissions and a user must grant the app all those permissions or they cannot install the app, versus an individual permission model, where a user can toggle individual permissions on or off for each app. There are several tradeoffs between these models:

- The all-or-nothing model is simpler for users in some sense, since they do not have to manage individual permissions, but it significantly restricts their ability to control their privacy.
- The all-or-nothing model is simpler for developers, since they not have to worry about how their app will perform with limited permissions.
- On many platforms that use an individual model, apps request access to a permission when they first use it, while all-or-nothing systems typically require users to agree to permissions when they app is first installed. Requesting permissions at use-time provides users with some context for how the permission is being used, but it can be annoying for users when the request interrupts a task.

Both all-or-nothing and individual models have been used in popular platforms. The iOS platform has used an individual permission model since it introduced permissions in iOS 6 [19]. Figure 2.1(c) shows



(a) An install-time permission screen in Android 4.4.

(b) A permission management screen in Android 6.

(c) A use-time permission request in iOS 10.

Figure 2.1: Install-time permission requests in Android 4.4, a permission management screen in Android 6, and a use-time permission request in iOS 10 (which also has a management screen similar to Android 6).

an iOS permission request. Google Chrome browser extensions [6] and early versions of Windows Phone employed an all-or-nothing approach, as did early versions of Android (shown in Figure 2.1(a)). Both scholars and popular writers expressed frustration with how this approach limited user control [31, 41]. Android 4.3 inadvertently exposed the permission toggling functionality [22], but removed it in Android 4.4.2, to some discontent [23]. Google pulled the feature because it caused apps to crash, as developers had not designed them to run with limited permissions (unlike most iOS apps, which have always had to contend with the possibility of not being granted a given permission). In Android 6, Google officially transitioned to an individual permission model (shown in Figure 2.1(b)), and current versions of Windows Phone also take the individual approach [5].

These variations between permission models affect how users manage their privacy. In Chapter 3 we will discuss some of these effects on users, and how they inform our choices about how to assist users.

Chapter 3

Privacy Decisions Facing Users

To better help users with privacy decisions, we need to understand what types of choices users actually make. At first blush, users face two types of privacy decisions: which apps to install and how to manage their apps' permissions after installation. If users need a *specific* app, managing permissions after installation is the only way for users to protect their privacy, and so they could benefit from a tool to assist them with that management, which would require privacy ratings for each permission. However, there may also be times where users can choose between similar apps, or they are using a platform which does not allow them to manage individual permissions (as discussed in Chapter 2), in which case a privacy-conscious store, with overall ratings for each app, would be helpful. To determine which tools to build, we studied whether users ever have a meaningful choice between different apps.

We posted Mechanical Turk surveys for 66 Android apps. For each app, we showed workers the app's description from Google Play, and asked whether workers thought that app was replaceable. If they thought it was, we asked if they could name an example substitute. If they thought the app could not be replaced, we asked why they felt it was unique. These questions are shown in Figure 3.1. We then asked several

Do you use ⟨app⟩? ▷ Yes: *No follow-up question* ▷ No: Do you use a similar app?
Do you think there are other apps that could be used in place of ⟨app⟩? ▷ Yes: Can you think of any examples of apps that could be used in place of ⟨app⟩? ▷ No: Why do you think that ⟨app⟩ is unique?

Figure 3.1: Classification survey questions for each app, with ⟨app⟩ replaced by the app's name. Workers were given the description of each app from the Play store.

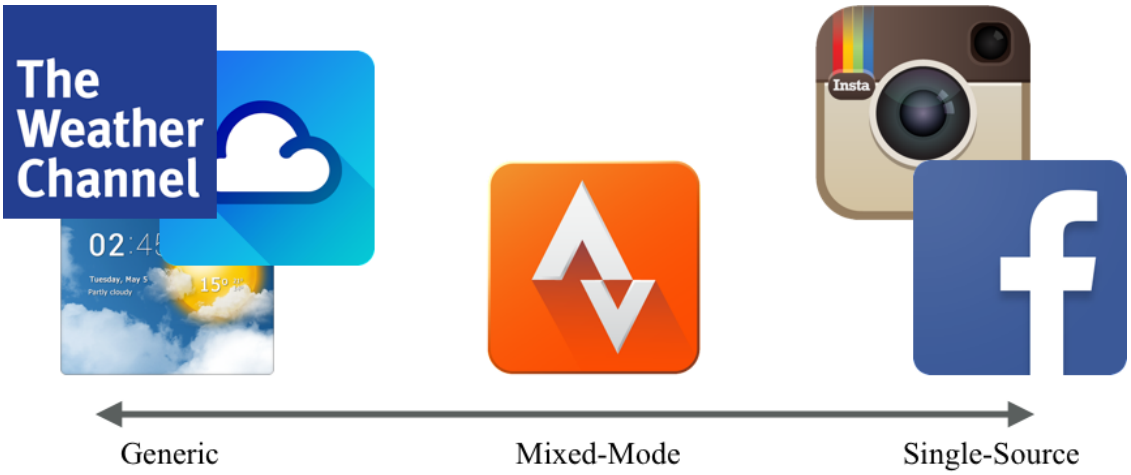


Figure 3.2: Apps fall along a spectrum of replaceability, from likely generic apps, like weather apps, to likely single-source, like Instagram or Facebook. Between these extremes are mixed-mode apps, whose classification depends on what features of the app the user needs.

demographic questions.

To select the 66 apps, we used the MarketBot scraper [3] to collect the descriptions of the top five apps in 11 of Google Play's categories, along with five white noise apps. We also chose six apps that were closely tied to a service external to the app, such as the Stop and Shop app, which is only useful at a physical Stop and Shop store. All of the apps had at least 100,000 installs, and only eight apps had less than 1M installs, suggesting that all the apps were interesting to a broad range of users. Figure 10.1 in the appendix shows the complete list of apps.

Each survey asked about three to five apps, and no survey contained two apps from the same category. We gathered 10 to 12 responses for each survey. Our workers were 61% male and 39% female, had an average age of 29, and 84% were from the United States and 16% were from India.

Apps varied significantly in their substitutability, (ANOVA, $p < 0.001$), indicating that some apps are interchangeable, while other apps provide unique functionality, tying users to that app. Rather than dividing clearly into replaceable or unique, however, we found that apps fall along a spectrum of substitutability, visualized in Figure 3.2. On one end are *single-source* apps, which offer unique functionality that cannot be replicated by a different app. Instagram is an example of a single-source app, as less than 20% of workers felt it could be replaced. On the other end of the spectrum are *generic* apps, such as Waze, which 100% of workers felt was replaceable. In the middle are *mixed-mode* apps, which can be either single-source or

generic depending on the user. For example, consider Strava, an app that allows users to track their physical activity and compete with friends. For users who only use the tracking features, it could be replaced by a similar app, such as MapMyRide. Other users might care deeply about the social features of Strava, and so other apps would not be an acceptable substitute.

Although there were not clear groupings of apps, some categories were more substitutable than others. For example, apps in the "social" category were considered, perhaps unsurprisingly, significantly less substitutable than apps in the "travel_and_local" category (Tukey's HSD, $p < 0.01$).

Ultimately, whether a given app is replaceable depends on the user, and therefore apps cannot be classified a priori. Overall, however, 30% of apps were considered "substitutable" by at least 75% of our workers, and 77% of apps were considered substitutable by at least 50% of workers. This indicates that users, whether they are aware of it or not, are making two distinct types of privacy choices: which apps to install (for generic apps), and how to manage apps' permissions after installation (for all apps, but most importantly single-source apps).

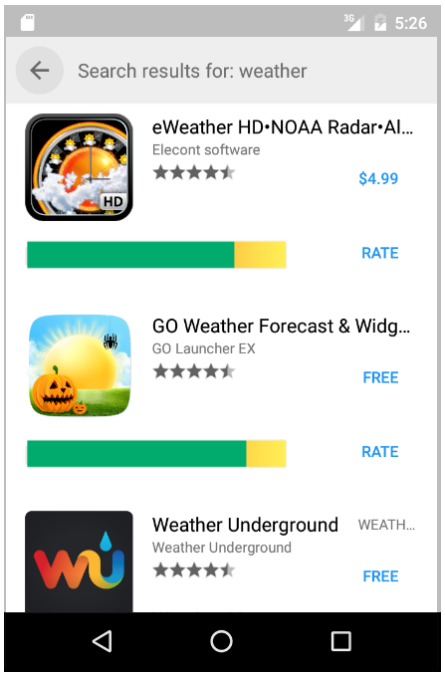
Chapter 4

Apps To Help Users Manage Privacy

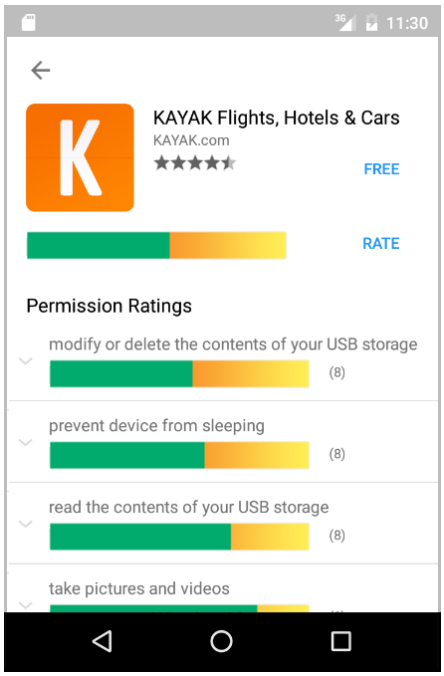
The two types of privacy decisions discussed in Chapter 3 require two approaches to assisting users. A privacy-aware marketplace would aid users with installation decisions by helping them find more privacy-respecting apps. A privacy assistant could help users manage their apps' permissions after they are installed on users' devices. We split these two approaches into two separate apps, the PerMission Store, and the PerMission Assistant[1]. Dividing the functionality into separate apps means that users who are only interested in one app are not required to accept the risks of both. In particular, the Assistant needs to access the list of apps the user has downloaded, information the Store does not need. This separation was practically useful in March of 2017, when Google Play updated its privacy requirements to classify "device information" as sensitive user data (an appropriate classification). Because the list of apps a user has installed is part of "device information," our assistant app was pulled for a privacy review, and required updates to comply with the new privacy requirements. Having two separate apps meant that the marketplace app was unaffected during this process.

Both apps already contain information for approximately 1500 Android apps from Google Play leaderboards, and are continuing to collect information for more apps.

[1] Both apps are available on the Google Play store, and can be found by going to `OnAPermission.org`.



(a) The search results page in the PerMission Store.



(b) The Kayak app page in the PerMission Store.

Figure 4.1: Screenshots of the PerMission Store.

4.1 The PerMission Store

The PerMission Store (shown in Figure 4.1) is designed to be a comprehensive app store, so, in addition to privacy ratings, it includes apps' description, screenshots, icon image, star rating, developer, category, and price from Google Play[2] and allows users to search and browse through apps, and rate permissions. There is one notable feature our store does not provide: it relies on the Play store to actually install apps. When users click to install an app in the PerMission Store, they are taken to that app's page in the Play store, where they can then install the app. Ideally, users would complete the entire process within our marketplace, but this would expose users to insecurity by requiring third-party downloads and by bypassing the malware protections in place in the Play store.

The PerMission Store displays privacy ratings at two levels: the permission-level and the app-level. Both levels of rating are represented with percentage bars developed via a series user interface design studies (Chapter 8). The permission-level ratings are comprised of both automated and human ratings as described in Section 5.3 and provide users with detailed information they can use to make privacy decisions. These

[2]Scraping the Play store, while not explicitly prohibited in the letter of the Terms of Service, is somewhat counter to their spirit. Integrating our store into Google Play would render this step unnecessary.

ratings are unique to a given app-permission combination, and so the same permission may have a different rating on different apps.

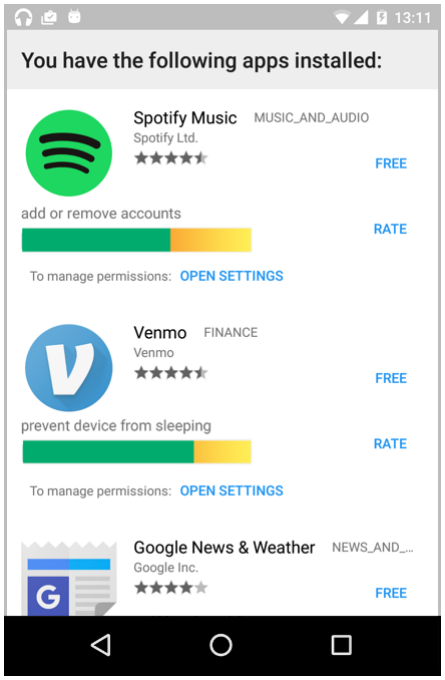
App-level ratings are calculated from permission-level ratings (Section 5.3), and serve several purposes. First, they are incorporated into the PerMission Store's ranking mechanism (discussed in Chapter 7), which is used to sort responses to user search queries, thus allowing the PerMission Store to promote more privacy-respecting apps. They also provide a broad privacy overview, making it easier for users to compare apps. Throughout the marketplace, an app's app-level privacy ratings are displayed next to its star rating from the Play store so that users can weigh both when choosing apps.

When users search or browse apps, they are shown tiles that display the apps' general information, like name, developer, app-level privacy rating, star rating, and price (see Figure 4.2(a)), as well as links to rate or install the app. If a user clicks one of these tiles they are taken to the app's page (an example of which is shown in Figure 4.1(b)), which has more detailed information like permission-level ratings and comments, and the app's description. The permissions are ordered worst-rated to best to ensure that users see the most worrisome permissions.

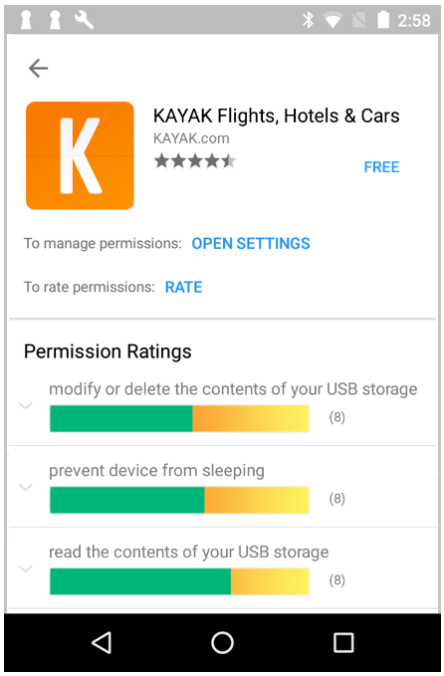
4.2 The PerMission Assistant

The PerMission Assistant (shown in Figure 4.2) helps users manage permissions for apps they have already installed. Because user time and attention is limited, the Assistant sorts a user's installed apps by their worst-rated permissions, which allows users to address the most concerning permissions first. It is thus useful for apps the user installed before the PerMission Store was available, and for single-source apps where the user cannot switch to a more privacy-respecting alternative. The Assistant allows users to run these apps within their own privacy limits. Because it relies on the ability to turn individual permissions off, the PerMission Assistant requires Android Marshmallow, while the PerMission Store can be used with any Android version.

The PerMission Assistant uses the same interface elements as the PerMission Store to display an app's permission ratings and provides a link to manage a given app's permissions (as seen in Figure 4.2(b)). Because we cannot actually edit other apps' settings, this link takes them to the app's page in their device's settings. This is, of course, a security necessity, because Android should not allow apps to adjust each others' permissions. However, it does mean that we cannot display privacy ratings on the actual adjustment screen



(a) The home page of the PerMission Assistant.



(b) The Kayak app page in the PerMission Assistant.

Figure 4.2: Screenshots of the PerMission Assistant.

in settings. Similarly, we cannot display ratings along with "just-in-time" permission requests that pop-up when an app requests a permission during use. The user can always look at the permission's ratings in the Assistant later, but ideally the ratings would be available at the time of the request. However, these dialogue boxes are a protected communication from the operating system, so we can not (and would not want to) inject rating information into them. Both of these issues could be solved if these ratings were incorporated into the Android infrastructure.

Chapter 5

Populating Privacy Information

The essential feature of our apps is privacy information, which we gather from two sources: an automated tool and human raters. As discussed in Chapter 4, our apps use both permission-level and app-level ratings. Since we cannot know, for a given app, whether a user will need permission-level ratings (to manage permissions) or app-level ratings (to choose between apps), we collect ratings for all apps at the permission level and compute an app-level rating from the permission-level ratings. Section 5.1 and Section 5.2 discuss collecting ratings from automated and human sources, and advantages and disadvantages of each. Section 5.3 discusses combining the human and automated ratings and calculating the app-level rating.

5.1 Automated Ratings

The research community has developed a number of systems that use automated techniques to provide privacy and security information about Android apps. Some attempt to identify malware apps [57, 58], while others detect worrisome permissions or suspicious handling of user data [17, 25, 52]. Chapter 9 offers further discussion of such systems.

These automated tools can provide objective, quantitative privacy information for a large number of apps at low cost. However, automated tools suffer from a number of short-comings. They are often difficult to use, even for sophisticated users (the author of this thesis was unable to get many of these tools to run). They provide little-to-no qualitative feedback, such as discomfort or confusion about permissions. Finally, many of these tools cannot consider the *context* of a permission (accessing contact data may be worrisome

for a flashlight app, but not a messaging app).

One of the few automated tools that offered a working installation is the DroidRisk system [51], which analyzes permission request patterns in both malware and benign apps to assign a risk score to each permission. (Because Android has added new permissions since the development of DroidRisk, the tool does not provide scores for all the current permissions.) Because it is a functional system which offers permission-level ratings, our apps incorporate DroidRisk ratings, but it should be noted that we are repurposing the tool, which was designed to detect malware rather than to rate legitimate apps.

Because we are using the DroidRisk ratings outside their intended purpose, and because they still lack important contextual and qualitative information like how users feel about a certain permission, our apps use the DroidRisk ratings primarily as a complement to the human ratings.

5.2 Human Ratings

To capture the full range of users' concerns, our apps employ user ratings and reviews, similar to the star ratings and text reviews in Google Play, along with the DroidRisk ratings. Of course, the average user is not a security expert, and thus may "mis-rate" a permission because they misunderstand its purpose. However, our apps aim to serve as a communication channel between users, developers, and the Android team, and "incorrect" ratings signal to developers that they are not adequately explaining their apps' permissions, and to the Android team that a permission is confusing or misleading. This is *vital* information, because *if users do not understand a permission, they cannot meaningfully consent to its use*, and therefore the permission system is failing in its primary objective. Because user ratings provide valuable information for other users, for app developers, and the Android team, our apps incorporate those ratings as a key source of permission information.

Bootstrapping Human Ratings Human ratings present a bootstrapping problem: Users will likely only use our apps if they contain ratings, but without ratings, the apps would struggle to gain the users necessary to rate apps. Our apps could initially rely only on automated ratings, but they would then suffer from the shortcomings of automated tools.

One option for seeding text reviews would be to mine the existing app reviews in Google Play, searching

for permission relevant text. However, Google Play makes it difficult to gather more than a sample of reviews for each app (40 per app, as of June 2016). The Play Store itself, should it ever integrate our apps' features, could leverage the complete database of existing reviews.

To offer human ratings right away, our apps use crowdsourced ratings from Mechanical Turk, which offers a cost-effective platform with a supporting body of academic research [39]. Although the Play store offers millions of apps, many of these apps are not at all widely used, so we have focused our seeding on popular apps by pulling from the Play Store's leaderboards (this is similar to the star ratings in the Play Store, where popular apps generally have numerous ratings while less popular apps may have few, if any). We have seeded our apps with crowdsourced ratings for over 1500 apps, and we are continuing to collect more. (The cost-effectiveness of Mechanical Turk enabled us to do this with a limited research budget.)

Crowdsourcing solves the bootstrapping problem, but raises concerns about whether workers take rating tasks seriously. (They might, for example, assign random ratings to finish the task as quickly as possible to maximize their income.) We thus performed a study to evaluate the quality of Mechanical Turk ratings.

We surveyed workers about 14 apps: Facebook, Gmail, Pandora, Angry Birds and ten weather apps, with 20-30 workers per app. For each app, we provided workers with its description and required permissions. We instructed workers to imagine that they were considering installing the given app and asked them, "Which, if any, of the permissions did you find unacceptable, and why?" They had to label each permission as either "acceptable" or "unacceptable," and could explain each rating in an optional text box.

We reviewed the text responses explaining the ratings. First, we found that more than 60% workers did provide explanations for their ratings, despite this being optional. Furthermore, their responses were relevant to the permissions being discussed, indicating that the workers performed the task seriously.

We also evaluated the quality of the binary ratings. This presented a challenge because, as ratings are essentially opinions, there is no ground truth against which to evaluate. We could measure agreement between workers with Fleiss's κ measure of inter-rater reliability, but low agreement would not necessarily mean that workers were negligent, since there could be valid disagreement. However, we *would* expect workers to agree on some of the permissions, particularly non-controversial ones, leading to a range of agreement across permissions. We computed κ scores for each permission and found that the scores ranged from -0.1 (significant disagreement) to 1.0 (total agreement). The scores aligned with our intuition about which permissions would be non-controversial. For example, `coarse-grained location` had $\kappa = 1.0$ for

all weather apps, which is unsurprising, as a weather app needs to fetch local conditions.

These findings suggest that Mechanical Turk is a viable method for seeding ratings for an initial corpus of apps. That said, we consider the crowdsourced ratings to be temporary. As we amass ratings from in-the-wild users, we will phase out crowdsourced ratings.

5.3 Merging Human and Automated Ratings

While having both human and automated ratings helps mitigate the shortcomings of each, it could be confusing and overwhelming for users to consider two ratings for every permission and to understand the distinctions between them. Thus, we merge each permission's human and automated ratings together, so that users can see questionable permissions at a glance.

Calculating the combined rating depends on whether the permission is in the DroidRisk corpus. If it is not, and thus does not have an automated rating, we take the average of its human ratings. If the permission *does* have an automated rating, we take a weighted average of the automated rating, denoted by *ar*, and the average of the human ratings, denoted by *hr*. The overall rating *PR* for a permission p is given by:

$$pr_p = (0.25 \times ar_p) + (0.75 \times hr_p) \tag{5.1}$$

where both *ar* and *hr* normalized to be between 0 and 1. Automated ratings are given a lower weight because they are a less nuanced metric than human ratings.

After computing a single rating for each permission, we have to calculate an overall privacy rating for each app. This app-level rating makes it easier for users to compare between multiple apps, and is necessary for ranking apps. An app that requires no permissions is given a privacy score of 1 (the best possible rating), because, from a permission standpoint, it is innocuous. For an app that does request permissions, we need to calculate an overall rating from its permissions' ratings. A naive approach would be to average the permissions' ratings (perhaps with some sort of weighting). However, an average would suffer a significant drawback: the aggregate rating would always be either equal to or *better* than the app's worst rated permission. As a result, an unscrupulous developer could hide a suspicious permission by requesting a large number of innocuous-seeming permissions. To avoid this, our marketplace uses an app's worst permission rating as the overall rating.

Chapter 6

The Effect of Brand Name

One element that could affect user trust of apps, and therefore influence permission ratings, is the brand behind the app. However, it's not clear *how* brand would affect ratings. On one hand, users could be more comfortable with permissions for a known brand app since they may feel the app will be better engineered and thus be more secure. On the other hand, they may feel that a large company may be more inclined to collect user data to better target ads. To explore these questions, we did a survey study examining how users would rate the permissions of otherwise identical apps with different branding.

We pulled the description and permission information of four apps from the Play store: Gmail, Waze, Pandora, and Instagram. To limit the length of the survey, we selected a subset of each app's permissions to include (between five and nine permissions per app). For each app, we created an "off-brand" version: MailMan (Gmail), ShortCuts (Waze), TuneUp (Pandora), and PictureIt (Instagram). The off-brand versions had the same description with the name of the app changed, and the same subset of permissions as their branded counterparts. We also removed any links from the description of both the on- and off-brand versions.

We posted survey tasks to Mechanical Turk for each of the eight conditions. Figure 6.1 shows a screenshot of the Gmail condition. We asked workers to read the description of the app and rate each one of its permissions on a 4-point Likert-type scale from "completely acceptable" to "completely unacceptable". For all conditions, the `Storage` permission was an attention check, where its description asked workers to leave a specific rating.

After workers rated the permissions, we asked them to answer several Likert-type questions about how

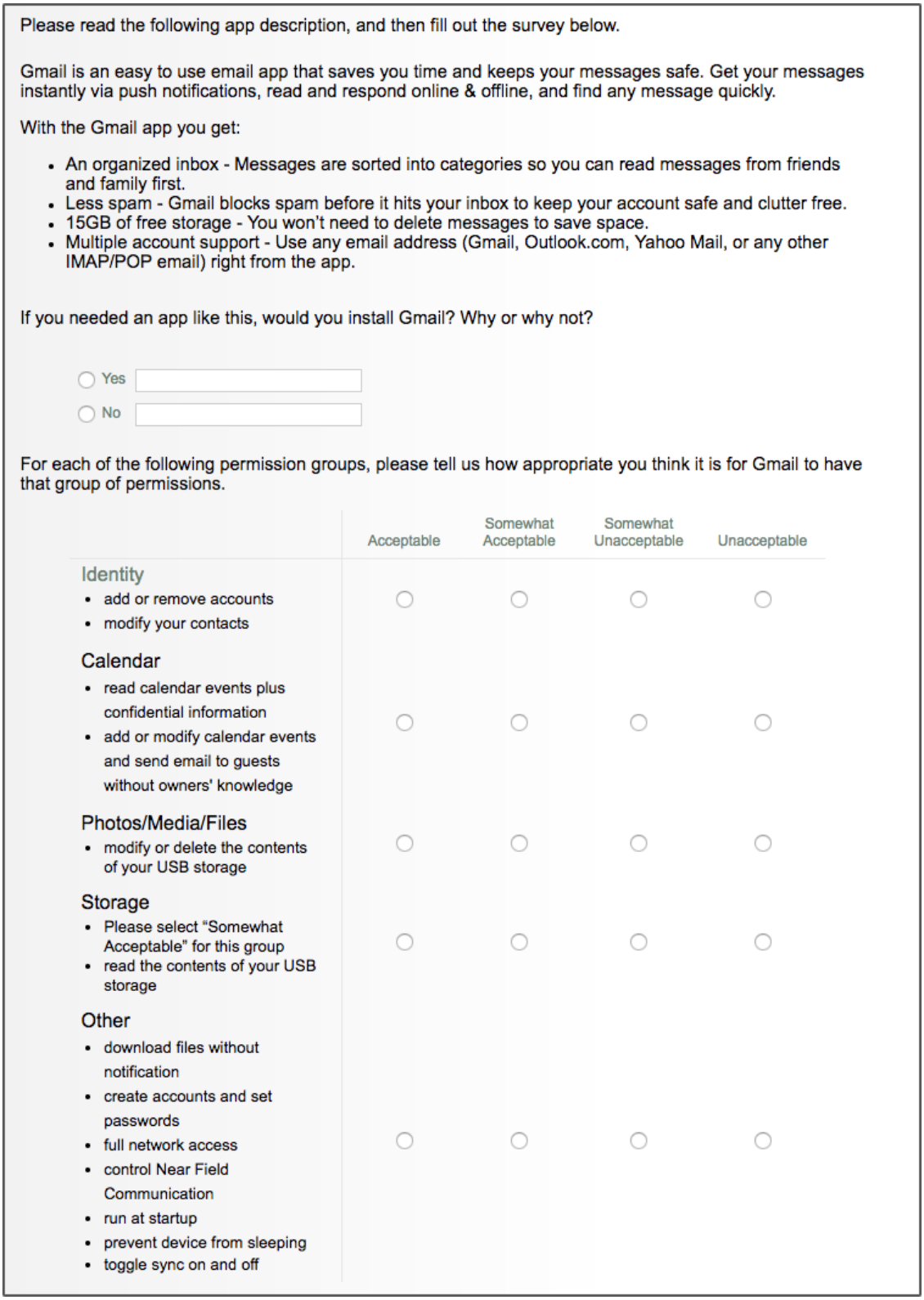


Figure 6.1: The rating section of the branding survey, showing the Gmail condition. Note that the first bullet under `Storage` asks workers to leave a "Somewhat Acceptable" rating for that permission.

trustworthy the app seemed, and how functional it seemed, to see if a known brand would influence perceptions of trustworthiness or quality, and if so whether those perceptions would affect permission ratings. The trustworthiness and quality questions were adapted from *Measuring Customer-Based Brand Equity* by Lassar et al. [35]. We also asked workers whether they had heard of the app before completing our survey. We eliminated any workers who either *had not* heard of a brand-name app, or who thought that they *had* heard of an off-brand app. The survey was a between-subjects design, so each worker saw only one app (and only one version of the app).

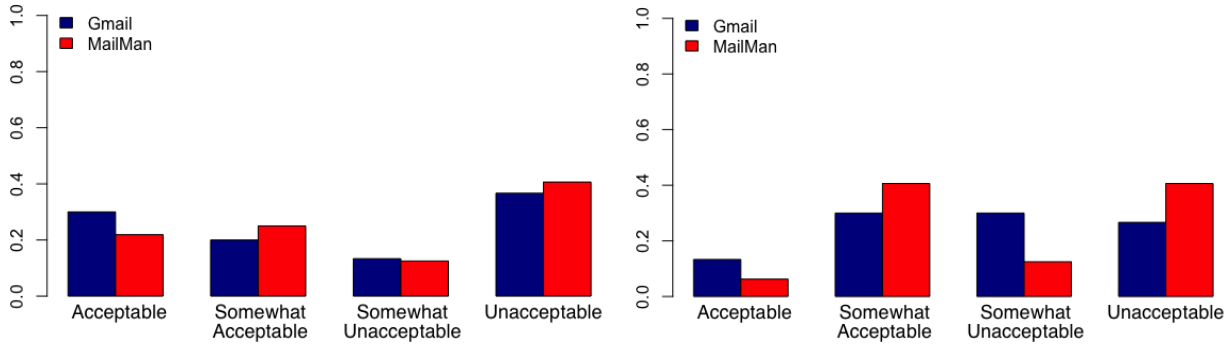
We chose to create off-brand versions of name-brand apps and gather survey responses for each, to more directly recreate the scenario of rating two similar apps, one familiar and the other unfamiliar. We chose this approach, rather than directly asking workers about their opinions on brand, to avoid the "privacy gulf": the gap between users' stated opinions about privacy and the actions they take (in this case, saying they feel one way about the importance of brand, but actually rating permissions in way that contradicts their stated beliefs).

Initially, we had a total of 274 respondents across all 8 conditions, with between 27 and 40 per condition. After eliminating responses based on the attention check, we were left with 233 total respondents (85% of the original number), with between 25 and 34 respondents per condition. When we removed respondents based on familiarity with the brand, we found that for all of the apps (both brand-name and off-brand) except for Waze, this check eliminated a small percentage of participants: between 3% and 16%, leaving between 21 and 32 respondents per app. However, for the app Waze, 67% of participants had not heard of the Waze app, and so there were only 9 respondents left.

After eliminating responses based on attention and brand-familiarity, we compared trustworthiness and quality ratings for each pair using an ANOVA analysis. We also compared the permission ratings between each pair of apps using an ANOVA analysis.

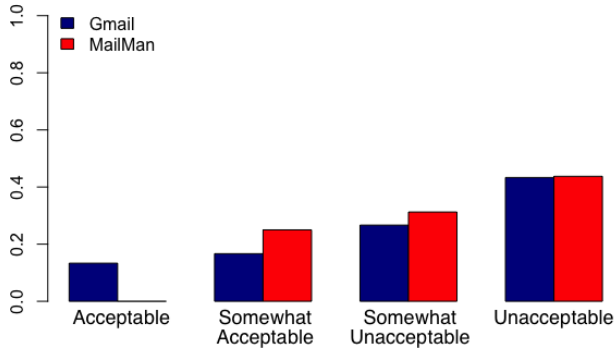
Results of trustworthiness analysis indicated that workers found Gmail ($M = 1.711$, $SD = 0.757$) to be significantly more trustworthy than MailMan ($M = 2.563$, $SD = 0.601$), with $F(1,60) = 24.21$, $p < 0.001$. Also, workers found Instagram ($M = 2.063$, $SD = 0.552$) to be more trustworthy than PictureIt ($M = 2.73$, $SD = 0.389$), with $F(1,51) = 23.14$, $p < 0.001$. The trustworthiness scores for Waze and ShortCuts, and Pandora and TuneUp were not significant.

For quality analysis, results showed workers found Pandora ($M = 2.032$, $SD = 0.759$) to be of higher

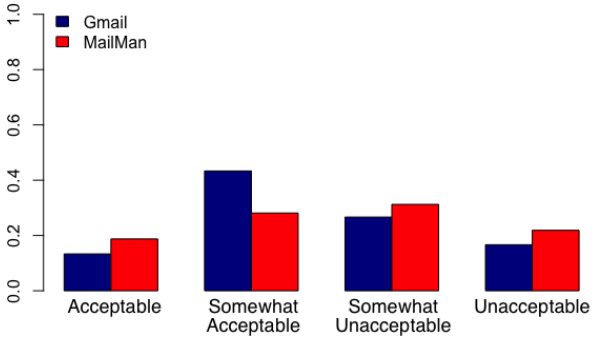


(a) The `Calendar` permission group.

(b) The `Photos/Media/Files` permission group.



(c) The `Other` permission group.



(d) The `Identity` permission group.

Figure 6.2: The ratings for each permission for the Gmail and MailMan apps (also shown in Figure 10.2 in the appendix). The ratings for other pairs of apps follow a similar pattern.

quality than TuneUp ($M = 2.32$, $SD = 0.605$), with $F(1,44) = 7.915$, $p = 0.007$. None of the other pairs showed a significant difference in quality.

Despite its effect on trustworthiness and quality, brand did not have any affect on the apps' permission ratings. For each pair, we found the brand of the app was not a significant factor in determining a permission ratings. Figure 6.2 shows the ratings for the Gmail and Mailman apps. The ratings for all four pairs of apps are shown in Figures 10.2, 10.3, 10.4, and 10.5 in the appendix.

These findings suggest that respondents do not consider the brand of an app when rating its permissions.

Chapter 7

Ranking Apps

While the privacy ratings can help users choose between apps, a privacy-conscious marketplace should also promote privacy-respecting apps so that users can find them in the first place. In particular, the marketplace should incorporate apps' privacy ratings into its search function so that apps with better privacy scores are ranked higher in results. However, the marketplace cannot simply sort results by privacy rating; users need apps that are functional and relevant to their needs, as well as privacy preserving.

One option would be to replicate the Play store's ranking for a given query and combine those rankings with our privacy ratings to sort apps. However, as discussed in Chapter 1, the Play store may rank apps in a way that is contrary to users' privacy interests, so integrating their ranking could undercut our goals. Also, the Play store's ranking method is opaque and could rely on privileged information, and so may be irreproducible. Thus, we need another way to incorporate functionality and relevancy.

Our marketplace uses apps' star ratings from the Play store as a proxy for functionality. These ratings are supplied by users, not by Google, and therefore do not present the same concerns as the Play store's ranking function.

To incorporate relevancy, we leverage our database of apps. The scraped app data are stored in a Postgres database. Postgres provides built-in text search that, given a search query, calculates a relevancy score for each record based on how often and where the query appears. Our marketplace searches against apps' title and description to get the relevancy score.

Given privacy, functionality, and relevancy information, we need compute a single ranking number because the marketplace ultimately needs a sort order for apps. Although we are building a privacy-conscious

marketplace, relevancy is the most important factor, followed by functionality, since users will not be satisfied with irrelevant or dysfunctional apps, no matter how privacy preserving. We use a weighted sum of all three components, so an app a's rank for a query q is defined by:

$$Rank_{aq} = r_{aq} + (0.25 \times fr_a) + (0.2 \times pr_a) \tag{7.1}$$

where r_{aq} is the relevancy score for app a on query q, fr_a is its functionality rating, and pr_a is the permission rating of its worst-rated permission (as defined in Equation 5.1), and r_{aq}, fr_a, and pr_a are normalized to be between 0 and 1. We arrived at these weights empirically by experimenting with different weights. Because relevance is the only component that depends on the search query, it carries more weight than functionality or privacy. If a user does not find an app they want after their initial query, and they try a second query, we want to return different results. Weighting fr_a and pr_a more heavily had the result that issuing similar but distinct queries (like "game" and "puzzle game") did not provide distinct results. Giving pr_a alone any more weight resulted in apps with poor functionality ratings appearing frequently in search results. Both of these effects could be frustrating to users.

Chapter 8

The Permission User Interface

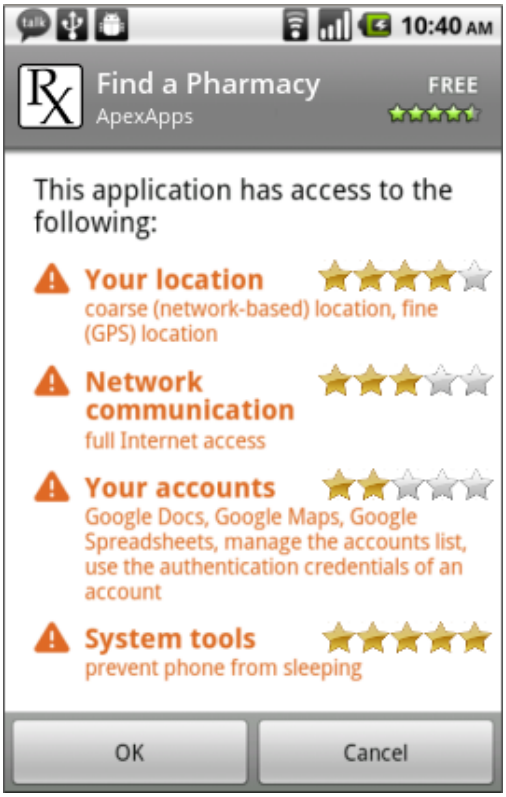


Figure 8.1: A prototype interface for permission ratings.

Because we want to communicate rating information to users, the interface for displaying the ratings is another critical component of our marketplace. The interface should help users understand the riskiness of individual permissions so they can make *informed* decisions without requiring significant effort. Ideally, it would be intuitive enough that users could understand it without too much direction. Figure 8.1 is an example of what such an interface might look like.

Designing such an interface proved surprisingly subtle. Our original designs, based on existing security metaphors, failed to convey the desired information. Indeed, we found that some of them *actively mislead users* (Section 8.1). We also unearthed some common patterns of interface confusion. In the end we found three designs that most subjects understood, and conducted a large user study to confirm this (Section 8.2). One of these designs will be used in the new marketplace.

8.1 Exploratory Interface Design

To find a functional interface, we designed several prototypes and leveraged Amazon's Mechanical Turk platform to give us rapid feedback on those prototypes.

Color in the Interfaces Several of our candidate interfaces use color to convey information. Although they all use colors distinguishable by viewers with red-green color vision deficiency (deployed apps are compatible with both red-green and blue-yellow color vision deficiency), they do lose meaning viewed in greyscale. Thus, recommend reading this section in color.

Methodology We explored each prototype with a survey on Mechanical Turk. These surveys were intended to expose broad conceptual problems in the interfaces, so we recruited only 10 to 12 subjects per interface. The surveys focused on two issues: how well subjects understood the purpose and meaning of the interface absent any explanation, and whether subjects understood where the ratings came from.

During each study, subjects were shown a mock-up of a candidate interface. Figure 8.1 is an example of such a mock-up. The mock-ups displayed the full permissions interface for a fictional app called Find a Pharmacy, which appeared to be developed by the (also fictional) company ApexApps. We chose a pharmacy locator app because it could pose a privacy risk to a user (if, for example, it stored a list of the user's medications for refill reminders), but would be unlikely to offend any subjects. Each mock-up used different iconography to present the user permission ratings (which were also fake), but the permissions and their rating values were the same or comparable across interfaces.

The mock-ups were presented as static images that were tall enough not to require scrolling. (Because the rating icons varied in size, the mock-ups varied in height.) This was both to ensure subjects did not miss any of the iconography by failing to scroll, and to avoid distraction induced by interaction.

Upon being presented with one of the mock-ups, subjects were asked to explain, in a free-response text box, what they thought the icons next to the permissions meant. Subjects were given no information about the purpose of the interface. The next page of the survey told them that the icons were privacy ratings and asked them to rate how clear this was from the interface, on a 4-point Likert-type scale.

We manually examined the text responses to identify conceptual problems with each interface, whereupon we either attempted to redesign the interface to address issues raised by subjects, or we decided the

interface was not viable and disqualified it. Using this process we eliminated all but three interfaces, which we evaluated in a larger study (Section 8.2).

To understand subjects' beliefs about the ratings' source, we asked whether they thought the ratings came from "other Android users", "independent security experts", "a review team at Google", or "don't know". I will discuss the outcome of this question before delving into the individual interfaces.

The Source of the Ratings If users are going to trust the ratings enough to use them, they are necessarily placing trust in the raters, so it is important that users understand the *source* of the ratings. We found that most of the interfaces failed to convey to subjects that the ratings were from other Android users. This is therefore something that should be considered in the design of the complete marketplace.

Stars A five-star system is possibly the most common iconography for user ratings, and is already in use in the Google Play store to display apps' overall functionality ratings. It is therefore a natural basis for experimentation.

Possibly due to the ubiquity of five-star ratings, subjects seemed to have preconceptions about the meaning and source of the ratings. This proved to be both an advantage and a disadvantage. On the positive side, subjects correctly understood the source of the ratings (other Android users), and that more stars corresponded to a better rating.

Unfortunately, subjects' association with stars as a *functionality* rating was *too* strong. Many subjects thought the ratings indicated how well the permissions' services worked. For example, some subjects thought the rating next to "Network Communication" showed the strength of the network signal.

In order for the star ratings to effectively communicate the meaning of the permission ratings, users would have to understand that the same icon on the same page had two different meanings (the app's functionality rating and the permission ratings). This potential for user confusion led us to eliminate this interface. However, it did inspire interfaces using privacy-relevant symbols rather than stars, with the intention of leveraging users' existing understanding of an out-of-five system while expressing that the ratings are about privacy.

Locks One symbol we used place of stars was locks, a common visual metaphor for protection. Our original lock design used yellow locks over a grey background: This design caused a number of misconceptions.

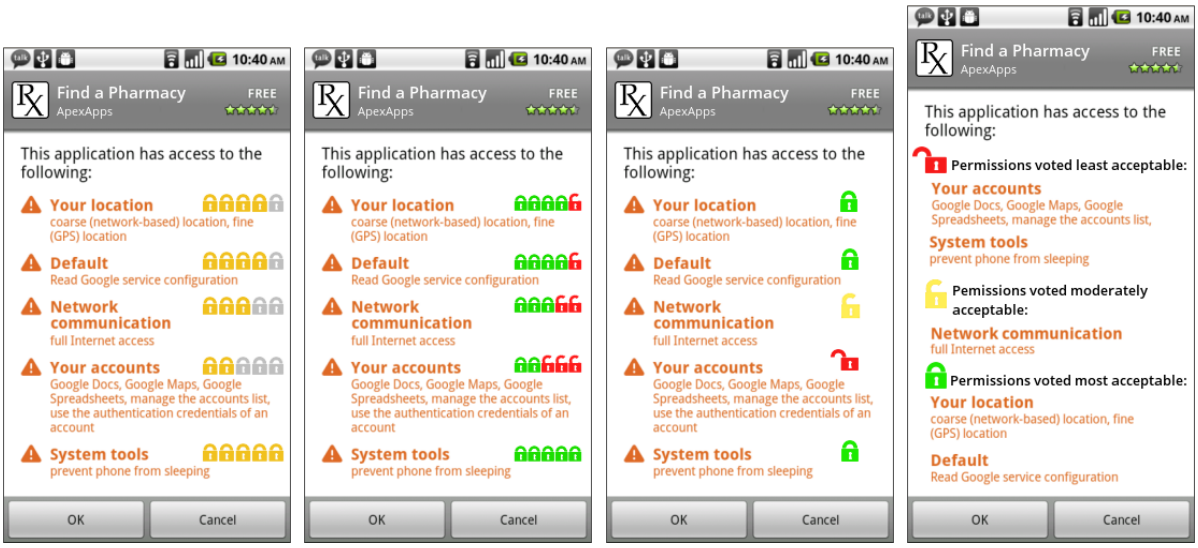


Figure 8.2: The lock interfaces

First, although most subjects understood that the locks were privacy ratings, some thought they meant that the permission's service was restricted. (This may stem from the practice by developers of using locks to mark features of an app that must be purchased or earned before they can be used.) Second, those subjects who *did* understand that the locks represented privacy ratings could not tell whether more yellow locks denoted a better or worse rating. This is troubling, because it would cause users to think the most dangerous permissions were the safest. We label this confusion, present in many interfaces, the *better-or-worse* phenomenon, and discuss it more at the end of this chapter.

The second lock interface, drawing from the traffic light interface (presented later in this chapter), tried to eliminate the better-or-worse phenomenon by using red and green locks. To further reinforce the message of privacy, the green locks were closed and the red locks open. We also hoped that using color would reduce the perception that the locks indicated restricted services (in which case *fewer* locks would be preferable). Though these changes helped curtail the better-or-worse phenomenon, they did not eliminate it entirely.

Because the better-or-worse phenomenon was at least partially caused by confusion about whether more or fewer icons was better, we replaced the out-of-five system with a single lock next each permission, and relied on color and open-ness to convey the rating: Using only red and green locks would have been too similar to the checkbox interface (discussed below), which had resulted in dangerous misunderstandings by subjects. To avoid this, the interface also used half-open yellow locks. This had the additional benefit of conveying more information than just red and green locks without adding much cognitive effort.

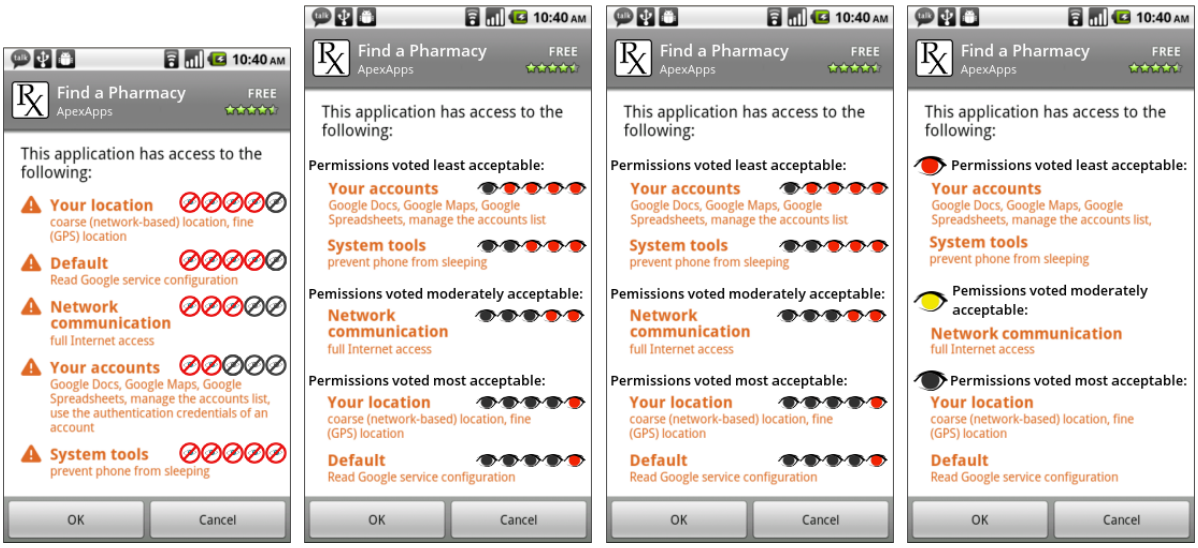
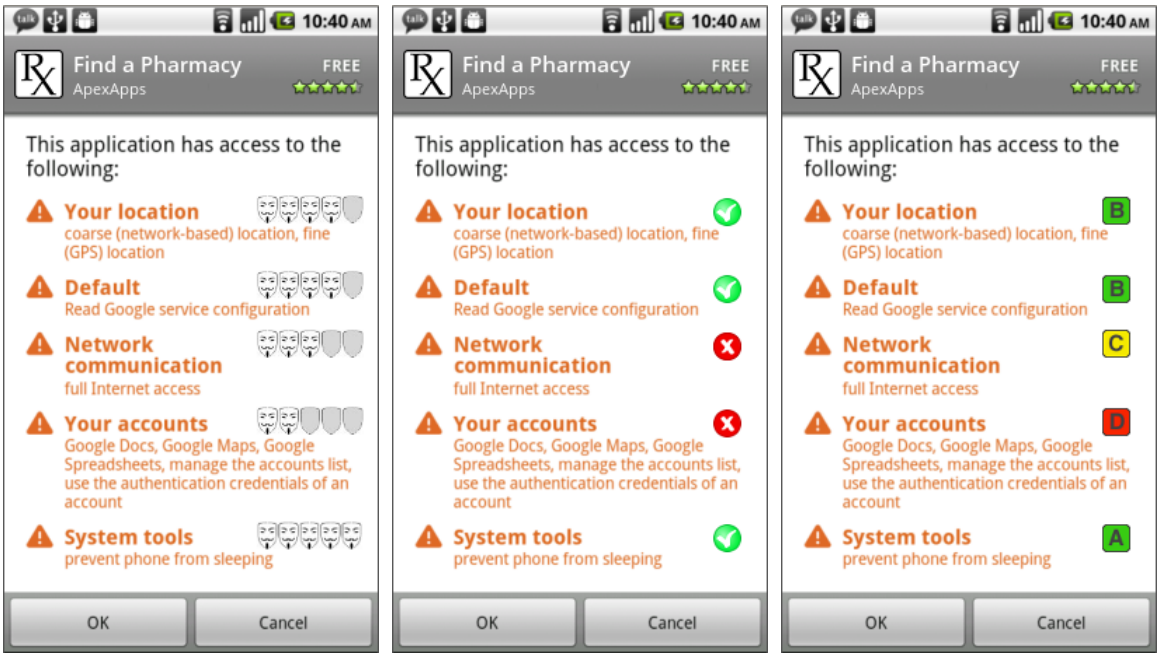


Figure 8.3: The eye interfaces

This redesign improved understanding, but some subjects still thought that the locks indicated inaccessible features. To further clarify the icons' meaning, we grouped permissions by rating and added explanatory text alongside the icons, drawing from the design of the first traffic light interface. (Additionally, we hoped introducing the word "voted" would also clarify the source of the ratings by emphasizing that they were an aggregate of community opinions.) The final lock interface was an improvement over its predecessors, but some subjects still thought the locks indicated availability. One subject said of the yellow lock, "I think it signifies that some features are unlocked but not all of them." Since locks performed worse than percentage bars (presented below) and traffic signs, we eliminated this interface family.

Eyes Continuing our exploration of other symbols in an out-of-five rating, this interface used eyes in the place of stars. Our first icon, which used a no-smoking-style circle-and-slash over an eye, proved too difficult to see at small scale. One subject stated that it "looks like a picture of a watch, so I would say it has something to do with time." We thus tried different-color eyes: The more dangerous a permission was, the more red eyes it had; the more benign it was, the more grey eyes it had. Additionally, the red centers had the appearance of a red recording light as seen on a camera.

Though subjects could now see the icon, this interface exhibited the better-or-worse phenomenon. One possible cause is that the grey eyes looked more like actual eyes, and so subjects thought that more grey eyes meant more surveillance.



(a) The mask interface. (b) The checkbox interface. (c) The grades interface.

Figure 8.4: The mask interface, the checkbox interface, and the grade interface.

We tried various redesigns such as grouping permissions by rating with a text header and introducing a yellow eye category. Though these changes helped, percentage bars and traffic signs were still better understood by subjects, so we disqualified this interface.

Guy Fawkes Masks We also explored out-of-five ratings using Guy Fawkes masks, which were popularized by the graphic novel *V for Vendetta* and its film adaptation, and have become a symbol for personal privacy and activism. Unfortunately, subjects felt the rating showed how well protected their information was from the government (possibly due to the "hacktivist" group Anonymous' adoption of the mask as a symbol). As this is not a protection a permissions system can provide and it is dangerous for an interface to suggest protections that do not exist, we eliminated all variations of this design.

Binary Checkboxes As we wanted to convey information without demanding much cognitive effort, we designed a simple interface in which each permission was given either a green checkmark indicating users approved of the permission or a red X indicating they did not approve. Unfortunately, we discovered a very significant confusion: in this interface the red X was meant to indicate a potentially invasive permission, but

subjects thought it meant that the given permission had been *disabled*. This is an extreme case of the better-or-worse phenomenon and is an alarming misconception. We therefore eliminated this interface without attempting to redesign it.

Grades Drawing on another iconography, this interface used letter grades to present the ratings. Typically used to rate students' academic performance, grades are also used in some non-educational settings (e.g., the New York City Department of Health restaurant inspection results). Unfortunately, most subjects thought the ratings were for the functionality of a permission's service. As this interface failed in its primary purpose, we eliminated it.

Percentage Bars Eschewing existing privacy and safety metaphors, this interface used rectangular bars to indicate the percentage of raters who considered a given permission to be acceptable. This style of rating conveys more information than the other interfaces, and therefore carries a greater risk of overwhelming a user. To mitigate this issue, the bars were colored red, yellow, or green depending on the permission's approval rating, giving a more obvious visual distinction between ratings: Subjects understood that the bars indicated privacy ratings, and this interface did not suffer from the better-or-worse phenomenon, due in part to the colors of the bars. One subject stated the bars rated the permissions from "most risky to the least, red being the highest and the green being generally safe".

Although the bars were effective, subjects' feedback on the traffic signs interface revealed a potential pitfall: their comments suggested that subjects perceived a green light as a signal to proceed without caution, which could encourage users to download an app without considering the permissions at all. We were concerned the green bars could have the same over-soothing effect.

To encourage caution in all cases, we modified the interface to use red, orange, and yellow bars. This interface had two variations. In both, more dangerous permissions had red bars and less dangerous permissions had yellow bars. In the first variant, the more dangerous a permission, the fuller its bar would be (showing the percentage of raters who deemed the permission *unacceptable*). These bars might look like  ,  , and  . In the second variation, the more dangerous a permission, the more empty its bar (showing the percentage of raters who deemed the permission *acceptable*). These bars would look like  ,  , and  .

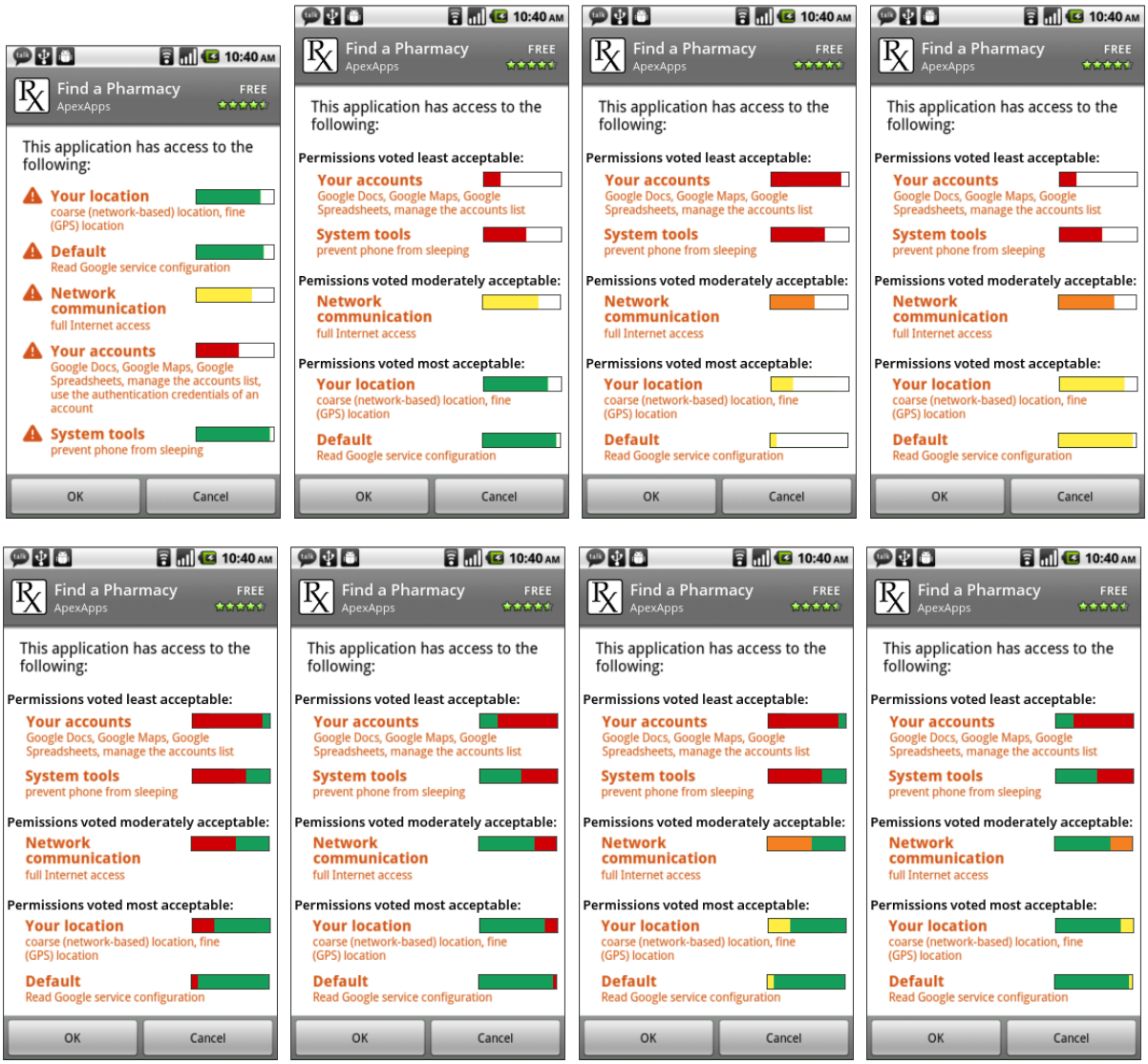


Figure 8.5: The bar interfaces

Both versions of this interface introduced the better-or-worse phenomenon. It is possible that, because all of the colors were "warning colors", the effectiveness of the color differentiation was diminished. Additionally these colors could cause warning fatigue after continuous use.

To avoid these problems, we introduced two-color bars. As before, each bar had some percentage of a warning color, (the percentage of raters who deemed the permission unacceptable for the app), however the rest of the bar was green, to clarify meaning and limit warning fatigue. There were four variants: The first two interfaces used only red and green (with two variants: red on the left or red on the right), so the goodness of a rating was indicated only by the ratio of red to green. Unfortunately, subjects thought these

bars were progress bars or ratings of the permission's service quality.

The second two interfaces used red, orange, and yellow along with the green, so the goodness of the permission was indicated both by the ratio of the warning color to green *and* by the warning color used. As with the red-green interfaces, one of the interfaces had the green on the left (so  ,  , and ), which we will call *G-ROY* bars, and the other had the warning color on the left (like  ,  , and ), which we will call *ROY-G* bars.

Unlike the red-green only bars, subjects still understood that the ratings were privacy related, and, unlike the warning-color only bars, they understood which ratings were better and which were worse. One subject said of the orange bar that "It means to me that feelings about this permission are mixed—about half of people think it is acceptable and half think it is not acceptable for this app to have that permission." Thus we subjected these interfaces to large-scale testing (Section 8.2).

Traffic Signs The final set of interfaces we designed used traffic markers, an iconography suggested by a subject from another interface.

The traffic marker interface split the permissions into three categories, with headers above each category. This interface successfully communicated that the ratings were related to privacy, but it exhibited a significant danger: the single green light gave subjects the sense that all the permissions in the "most acceptable" category were completely safe and did not need to be examined at all, which is not necessarily the intended meaning. Additionally, this interface could be unsuitable for users with color vision deficiency.

To address color vision deficiency issues, we tried a variation using position (as real traffic lights do). However, it still did not address the problem of an overly-soothing green light.

Rather than simply changing the colors of the lights (which could look jarringly different from actual traffic lights and thus confuse users), the next interface used traffic *signs*: a red octagon (mimicking a stop sign), an orange diamond, and a yellow circle. The colors would be sparser in the interface (as they were only by the section headers), so warning fatigue was less of a concern than in the percentage bars. This interface was well understood by subjects so we included it in the large-scale testing (Section 8.2).

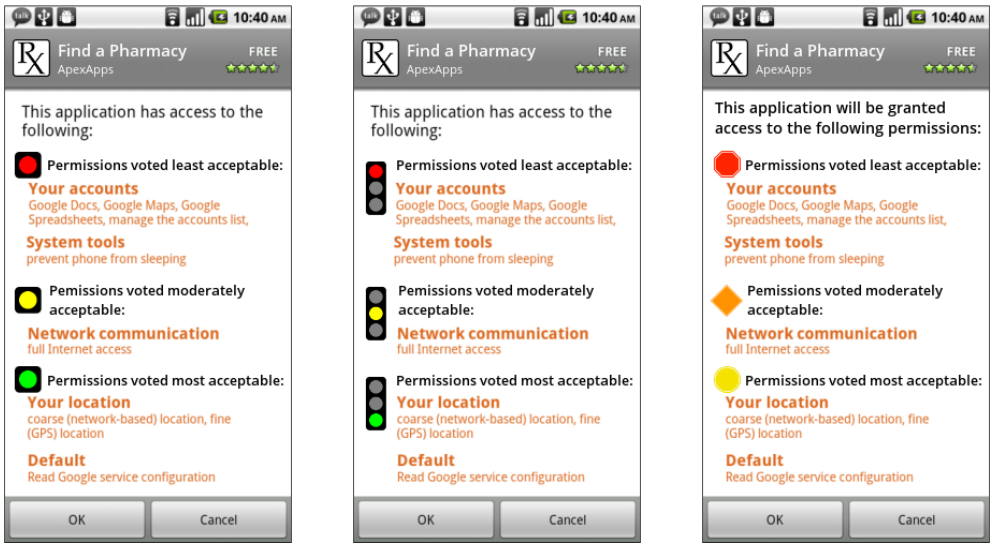


Figure 8.6: The traffic interfaces

Common Findings and Observations These studies exposed two issues that arose in multiple interfaces. First, because Android is used in a range of cultures, some metaphors may not be familiar or applicable to all users. For example, some countries do not use a letter grades in their schools. Of our interfaces, only the percentage bars do not rely on an existing metaphor and so avoid this particular confusion.

The second common issue was the better-or-worse phenomenon, wherein the more negative a rating is, the more positive subjects interpreted it to be. The net effect of this is alarming: The most dangerous permissions appear to be the most harmless! This problem is most troubling in the checkbox interface. There, dangerous permissions were indicated by a red X, but subjects thought the X meant that the permissions had been disabled, and therefore were *completely innocuous*. Because of this phenomenon's dangerous nature, it greatly influenced our design decisions, and our selection of interfaces to study further.

8.2 Large Scale Interface Evaluation

Small-scale testing allowed us to eliminate all but three interfaces. To further validate these three, we carried out a large scale evaluation.

Methodology As with the smaller studies, we posted surveys on Mechanical Turk. We had 311 subjects for the traffic signs interface, 365 subjects for the G-ROY bar interface, and 83 subjects for the ROY-G bar interface. The surveys explored four issues.

Two are the same as before: how well subjects understood the meaning of the interface absent other cues, and whether subjects understood the ratings' source. For these, we used the same prompts and mock-ups as for the small-scale studies (Chapter 8.1).

In addition, we asked subjects how much they would trust ratings from each of the three possible sources ("other Android users", "independent security experts", and "a review team at Google"). For each source, subjects had to select either "I would not trust them at all", "I would trust them somewhat", or "I would trust them completely".

Finally, we examined whether subjects would consider these ratings useful for different types of users. Specifically, we asked them to provide "yes"/"no" answers for whether they would use such a system for themselves, recommend the system for use by a parent (someone who might need assistance with technical decisions), and recommend it for use by a teenager (someone for whom they might be responsible).

I will first discuss how well subjects understood the interface, than I will discuss the perception of utility of these ratings for different populations. Finally I will discuss whether subjects understood the source of the ratings and how much they would trust each source.

The Meaning of the Ratings We had two types of data to evaluate how well subjects understood the meaning of the interface: text answers to the free-response question, and the Likert scale data.

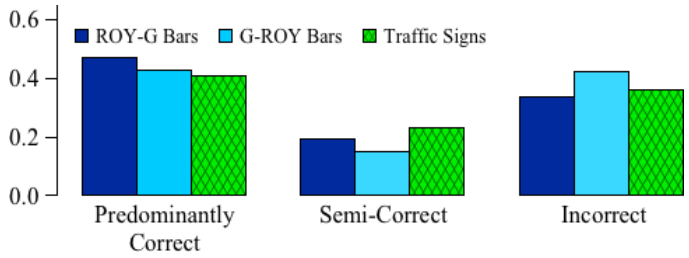


Figure 8.7: Percentage of subjects in each interpretation category.

To evaluate the text data we manually coded the correctness of the interpretation of the interface for each of the responses.

To classify, we used a rubric that was revised until we obtained an inter-coder reliability score (κ) of 0.835. The rubric is as follows:

Predominantly Correct Interpretation User understands, for all three ratings, that the ratings signify the acceptability of each permission or believes that the rating signifies the potential harm that could be caused by each permission, and correctly interprets the order of ratings from positive to negative. Users who correctly identify which of two ratings is more positive, but do not explain their choice

further, fall in to this category.

Semi-Correct Interpretation User understands that the ratings are privacy related, but does not understand or mis-understands exactly what they signify (e.g., they may think it signifies how often the app uses a service, or the "level" of access the app has to the rated service). Users who understand that the ratings are privacy-related, but cannot correctly interpret the order of ratings from positive to negative fall in to this category. Users who think the ratings signify how much access an app has to the permission fall in to this category. Additionally, users who believe that the ratings signify the percentage of behaviors pertaining to a given permission that are acceptable (e.g. half of the network communications by the app are acceptable) fall in to this category.

Incorrect Interpretation User does not understand that the ratings are privacy related (e.g., they may think the ratings have to do with power usage), or that they are ratings at all.

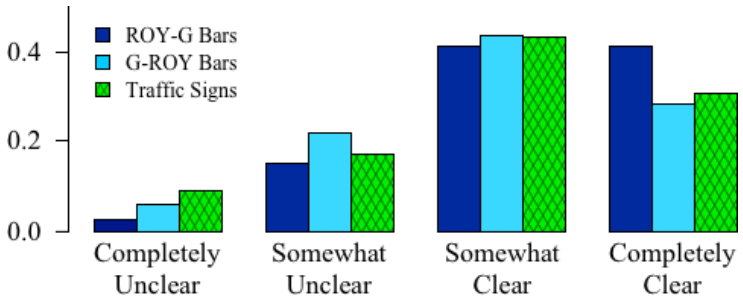


Figure 8.8: Subjects' responses to the Likert-type question asking users whether they thought it was clear that the icons represented privacy ratings.

Fig. 8.7 summarizes the percentages of subjects in each class for each interface.

Broadly classifying both correct and semi-correct interpretations as understanding the interface, all three interfaces were understood by over 50% of subjects. In all three cases, a chi-squared goodness-of-fit test showed the interface performed better than chance. For the G-ROY bar interface, 58% of subjects understood the interface, $\chi^2(1, N = 365) = 8.29, p = 0.004$. The traffic sign interface was understood by 64% of subjects, $\chi^2(1, N = 311) = 24.34, p < 0.001$. For ROY-G bars, 66% of subjects understood the interface, $\chi^2(1, N = 83) = 8.78, p = 0.003$.

Communicating Rating Information Without Context Note that these subjects had been given no explanation for the interface, so these results represent a worst-case baseline. This would likely not occur in

the privacy-focused app store where users would have context for the ratings' meaning. However, ideally our privacy features would be incorporated into a general purpose marketplace, where users may not have the context to cue them to the meaning of the ratings. In this case, the ratings would need to communicate their meaning absent the context.

After subjects answered the free-response questions, we informed them that the icons were privacy ratings. We asked them to rate whether this was clear from the interface, on a 4-point Likert-type scale from "completely unclear" (a value of 1) to "completely clear" (a value of 4). The responses are shown in Fig. 8.8. The ROY-G bar interface performed the best, with a mean of 3.2, and a chi-squared goodness-of-fit test showed the results to be significantly different from chance, $\chi^2(3, N = 83) = 36.3, p < 0.001$. Traffic signs had a mean of 3, and the chi-squared test showed the results to be significantly different than chance, $\chi^2(3, N = 311) = 83.32, p < 0.001$ The G-ROY bars had a mean of 2.9, and again, a chi-squared test showed a significant difference from chance, $\chi^2(3, N = 365) = 105.38, p < 0.001$

Likelihood of Recommending the System To determine whether subjects felt the ratings would be useful for different audiences, we asked if they would personally use the ratings, and if they would recommend them to a parent or teenager. Subjects' responses did not differ significantly between interfaces, so we consider them in aggregate. Overall, 75% of subjects said they would use the ratings for themselves, 74% would use them for a teenager, and 72% would use them for a parent. These results suggest there is a user base for these ratings.

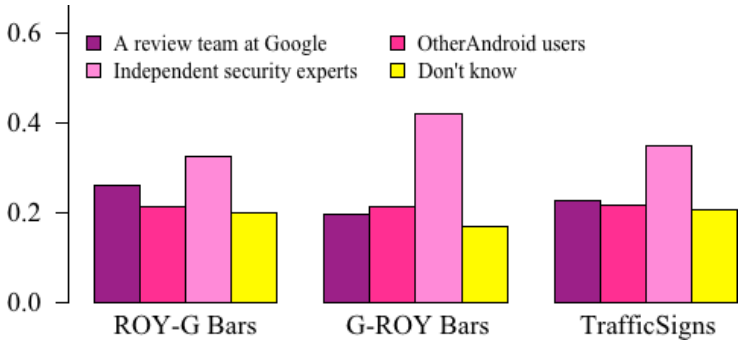


Figure 8.9: Subjects' beliefs about the source of the ratings.

The Source of Permission Ratings As in the small-scale studies, we investigated whether subjects understood that the ratings were from other Android users. Subjects' beliefs are shown in Fig. 8.9. During this large-scale study, a plurality of subjects understood that they were from other Android users, but there was still confusion. Although it would be ideal, it

may not be possible to convey the source of the ratings solely through an interface.

Overall, this study suggests that users are concerned about their privacy but currently lack the tools or expertise to control their own data and resources. Our marketplace's privacy ratings of permissions will provide users with a mechanism to make informed decisions about apps.

Chapter 9

Related Work

There are a number of "permission manager" apps on Google Play, many of which simply reorganize the information provided in the Android settings, and do not offer any additional privacy information. Some highlight "risky" apps, but it is not clear how they are calculating risk [10, 11]. Many appear to use the number of permissions a given app requests, which is an unreliable metric. There are also managers that remove other apps' permissions by altering the apps' APKs [8], or require root access to disable permissions [9], which are significant threat vectors in their own right and do not actually help users make privacy choices (and are of limited use since the release of Android Marshmallow, where permission toggling is a built-in feature). None of these tools provides the structured permission ratings and reviews available in our PerMission Assistant.

Almuhimedi et al. [13] show that a permission manager can be helpful to users in managing their privacy. Liu et al. [37] present a personalized privacy assistant (PPA) that engages users in a dialogue to determine a privacy profile for the user, which the manager then employs to suggest permission settings to the user. Although similar in concept, by focusing on publicly viewable ratings, our system can both let users explore how *other* users understand permissions, and serve as a channel of communication amongst users, developers, and the Android team. Our Assistant could be incorporated with the PPA to provide a more complete tool.

Highlighting the value of privacy information in the marketplace, researchers such as Felt et al. [27] have found that smartphone users take privacy risks seriously. In a study conducted while Android was still using an all-or-nothing permission model, Wijesekera et al. [54] found that at least 80% of respondents

would have liked to block at least one permission request, indicating they are concerned about their privacy. However, Chin et al. [18] show that although smartphone users are careful about performing certain tasks, they engage in risky behavior when it comes to installing apps, suggesting that users could benefit from a more privacy-conscious marketplace. Tsai et al. [50] built a search engine annotated with privacy scores for the merchants. They found that users are more likely to purchase products from sellers with higher privacy scores, demonstrating that offering privacy information during the search process can affect user decisions. Tan et al. [48] show that when iOS developers provide explanations of their apps' permission requests, users are more likely to approve the requests. This indicates that users want to understand how permissions will be used, and that it is in developers' interest to provide this information.

Tian et al. [49] use app reviews to give users more privacy information, showing that user reviews can help users make privacy decisions. However, they focus on the consequences of app updates, rather than installing new apps or managing current apps. Additionally, they draw from existing reviews, rather than gathering privacy-specific reviews.

There are systems that use automated approaches to detect misbehavior or privacy risks in apps (such as Chin et al. [17], Enck et al. [25], Sarma et al. [46], and Wei and Lie [52]), to flag dangerous permissions (such as Wang et al. [51] and Pandita et al. [42]), or to detect malware (like Zhou et al. [57], Zhou et al. [58], and others). All of these systems generate information that could be employed in a privacy-centric marketplace to rank apps and inform users about privacy. Yu et al. [56] and Rosen et al. [45] use API and method calls to generate privacy policies for Android apps, and to highlight privacy-relevant app behavior, respectively, but neither system connects particular behaviors with the permissions that enable them. If developers or Android were to provide this information, our PerMission Assistant could incorporate these tools to help users decide which permissions to enable or disable. Ayyavu and Jensen [14] find that user feedback (such as ratings) and heuristic-based automated tools are complementary.

Papamartzivanos et al. [43] analyze smartphone usage patterns across users to find privacy leaks in apps. Lin et al. [36] and Yang et al. [55] use information gathered via crowdsourcing to find unexpected permissions and improve user understanding of Android permissions. These systems aggregate crowd feedback into observations about apps, rather than providing a direct channel of communication for users and developers. Burguera et al. [16] also take a crowd-based approach to app security. Unlike our work, they use the crowd to collect traces of app behavior to detect malware, rather than gathering direct feedback from users

on permission use in legitimate apps.

Kelley et al. [33] find that using a "nutrition label" format for privacy policies helped users better understand the policy. Egelman et al. [24] use crowdsourcing to evaluate user comprehension of privacy icons for ubiquitous computing environments. These works demonstrate how an interface can help users better understand privacy, but their icons are intended for different uses.

Bibliography

[1] Family hub refrigerator. Accessed May 2017. http://www.samsung.com/us/explore/family-hub-refrigerator/.

[2] Google Play Store: number of apps 2009-2016 | statistic. Accessed Feb. 2016. https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/.

[3] Google Play Android app store scraper. Accessed Apr. 2016. https://www.github.com/chadrem/market_bot.

[4] HomeOS: Enabling smarter homes for everyone - Microsoft Research. Accessed May 2017. https://www.microsoft.com/en-us/research/project/homeos-enabling-smarter-homes-for-everyone/.

[5] App permissions explained - what are they, how do they work, and should you really care? Written Jan. 2016. https://www.dbbest.com/blog/app-permissions-explained/. Accessed: Mar. 2017.

[6] Permissions requested by apps and extensions - Chrome web store help. Accessed Mar. 2017. https://support.google.com/chrome_webstore/answer/186213?hl=en.

[7] Android permissions | help center. Accessed Apr. 2016. https://help.pinterest.com/en/articles/android-permissions.

[8] Apk permission remover - Android apps on Google Play. Accessed Apr. 2016. https://play.google.com/store/apps/details?id=com.gmail.heagoo.apkpermremover.

[9] Fix permissions - Android apps on Google Play. Accessed Apr. 2016. https://play.google.com/store/apps/details?id=com.stericson.permissionfix.

[10] MyPermissions privacy cleaner - Android apps on Google Play. Accessed Apr. 2016. https://play.google.com/store/apps/details?id=com.mypermissions.mypermissions.

[11] PermissionDog - Android apps on Google Play. Accessed Apr. 2016. https://play.google.com/store/apps/details?id=com.PermissioDog.

[12] This is your brain on Uber. Written May 2016. http://www.npr.org/2016/05/17/478266839/this-is-your-brain-on-uber. Accessed: Jan. 2017.

[13] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal. Your location has been shared 5,398 times!: A field study on mobile app privacy nudging. In *Conference on Human Factors in Computing Systems*, 2015.

[14] P. Ayyavu and C. Jensen. Integrating user feedback with heuristic security and privacy management systems. In *Conference on Human Factors in Computing Systems*, 2011.

[15] J. Bernstein. You should probably check your Pokemon Go privacy settings - BuzzFeed News. Written July 2016. https://www.buzzfeed.com/josephbernstein/heres-all-the-data-pokemon-go-is-collecting-from-your-phone?utm_term=.ceMJPj2k7#.hnVZpx89n. Accessed: Apr. 2017.

[16] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani. Crowdroid: Behavior-based malware detection system for Android. In *Security and Privacy in Smartphones and Mobile Devices*, 2011.

[17] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner. Analyzing inter-application communication in Android. In *Mobile Systems, Applications, and Services*, 2011.

[18] E. Chin, A. P. Felt, V. Sekar, and D. Wagner. Measuring user confidence in smartphone security and privacy. In *Symposium on Usable Privacy and Security*, 2012.

[19] J. Cipriani. How to control your privacy settings on iOS 6 - CNET. Written Sept. 2012. https://www.cnet.com/how-to/how-to-control-your-privacy-settings-on-ios-6/. Accessed: May. 2017.

[20] G. Cluley. IT manager has bikes stolen after cycling app reveals his address. Written Dec. 2015. https://www.welivesecurity.com/2015/12/22/manager-bikes-stolen-cycling-app-reveals-home-address/. Accessed: Apr. 2017.

[21] J. Cox. Hack brief: Malware sneaks into the Chinese iOS App Store | WIRED. Written Sept. 2015. https://www.wired.com/2015/09/hack-brief-malware-sneaks-chinese-ios-app-store/. Accessed: Apr. 2017.

[22] P. Eckersley. Awesome privacy tools in Android 4.3+. Accessed Mar. 2015. eff.org/deeplinks/2013/11/awesome-privacy-features-android-43.

[23] P. Eckersley. Google removes vital privacy feature from Android, claiming its release was accidental. Accessed Mar. 2015. eff.org/deeplinks/2013/12/google-removes-vital-privacy-features-android-shortly-after-adding-them.

[24] S. Egelman, R. Kannavara, and R. Chow. Is this thing on?: Crowdsourcing privacy indicators for ubiquitous sensing platforms. In *ACM Conference on Human Factors in Computing Systems*, 2015.

[25] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. In *Operating Systems Design and Implementation*, 2010.

[26] D. Etherington. GM's new SDK for in-car infotainment apps offers access to nearly 400 data points | TechCrunch. Written Jan. 2017. https://techcrunch.com/2017/01/26/gms-new-sdk-for-in-car-infotainment-apps-offers-access-to-nearly-400-data-points/. Accessed: May. 2017.

[27] A. P. Felt, S. Egelman, and D. Wagner. I've got 99 problems, but vibration ain't one: A survey of smartphone users' concerns. In *Security and Privacy in Smartphones and Mobile Devices*, 2012.

[28] D. Frommer. The smart TV app revolution - Business Insider. Written Oct. 2013. http://www.businessinsider.com/the-smart-tv-app-revolution-2013-10?utm_source=House&utm_term=RR&utm_campaign=RR. Accessed: Apr. 2017.

[29] A. Gell. The not-so-surprising survival of Foursquare - the New Yorker. Written Mar. 2017. http://www.newyorker.com/business/currency/the-not-so-surprising-survival-of-foursquare. Accessed: Apr. 2017.

[30] D. Goodin. Golden state warriors android app constantly listens to nearby audio, fan says. Written Sept. 2016. https://arstechnica.com/tech-policy/2016/09/golden-state-warriors-android-app-constantly-listens-to-nearby-audio-fan-says/. Accessed: Jan. 2017.

[31] C. Hoffman. iOS has app permissions, too: And they're arguably better than Android's. Written Dec. 2013. https://www.howtogeek.com/177711/ios-has-app-permissions-too-and-theyre-arguably-better-than-androids/. Accessed: May. 2017.

[32] J. Kahn. Apple redesigns Siri with new features in iOS 7, introduces iOS in the car. Written June 2013. https://9to5mac.com/2013/06/10/apple-redesigns-siri-with-new-features-in-ios-7-introduces-ios-in-the-car/. Accessed: Jan. 2014.

[33] P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder. A "nutrition label" for privacy. In *Symposium on Usable Privacy and Security*, 2009.

[34] J. Laird. Google's Android OS is mating with cars at CES, promising big things for your ride. Accessed Jan. 2014. techradar.com/news/car-tech/google-s-android-os-is-mating-with-cars-at-ces-promising-big-things-for-your-ride-1212393.

[35] W. Lassar, B. Mittal, and A. Sharma. Measuring customer-based brand equity. In *Journal of Consumer Marketing*, volume 12, 1995.

[36] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *Mobile Ubiquitous Computing, Systems, Services and Technologies*, 2012.

[37] B. Liu, M. S. Andersen, F. Schaub, H. Almuhimedi, S. A. Zhang, N. Sadeh, Y. Agarwal, and A. Acquisti. Follow my recommendations: A personalized privacy assistant for mobile app permissions. In *Symposium on Usable Privacy and Security*, 2016.

[38] F. Manjoo. Clearing out the app stores: Government censorship made easier. Written Jan. 2017. https://mobile.nytimes.com/2017/01/18/technology/clearing-out-the-app-stores-government-censorship-made-easier.html. Accessed: Apr. 2017.

[39] W. Mason and S. Suri. Conducting behavioral research on Amazon's Mechanical Turk. *Behavior Research Methods*, 44, 2012.

[40] L. Mirani. The amount most people are willing to pay for an app is $0-until they've actually downloaded it - Quartz. Written Sept. 2013. https://qz.com/129699/the-amount-most-people-are-willing-to-pay-for-an-app-is-0-until-theyve-actually-downloaded-it/. Accessed: Apr. 2017.

[41] M. Nauman, S. Khan, and X. Zhang. Apex: Extending Android permission model and enforcement with user-defined runtime constraints. In *ACM Symposium on Information, Computer and Communications Security*, 2010.

[42] R. Pandita, X. Xiao, W. Yang, W. Enck, and T. Xie. WHYPER: Towards automating risk assessment of mobile applications. In *USENIX Conference on Security*, 2013.

[43] D. Papamartzivanos, D. Damopoulos, and G. Kambourakis. A cloud-based architecture to crowdsource mobile app privacy leaks. In *Panhellenic Conference on Informatics*, 2014.

[44] K. Pratap. Android M: Top new features in the next major android release. Accessed June 2015. `gadgets.ndtv.com/mobiles/features/android-m-top-new-features-in-the-next-major-android-release-697502`.

[45] S. Rosen, Z. Qian, and Z. M. Mao. AppProfiler: A flexible method of exposing privacy-related behavior in Android applications to end users. In *Conference on Data and Application Security and Privacy*, 2013.

[46] B. P. Sarma, N. Li, C. Gates, R. Potharaju, C. Nita-Rotaru, and I. Molloy. Android permissions: A perspective combining risks and benefits. In *Symposium on Access Control Models and Technologies*, 2012.

[47] J. Skillings. Overexposed: Snapchat user info from 4.6M accounts - CNET. Written Jan. 2014. `https://www.cnet.com/news/overexposed-snapchat-user-info-from-4-6m-accounts/`. Accessed: Apr. 2017.

[48] J. Tan, K. Nguyen, M. Theodorides, H. Negrón-Arroyo, C. Thompson, S. Egelman, and D. Wagner. The effect of developer-specified explanations for permission requests on smartphone user behavior. In *Conference on Human Factors in Computing Systems*, 2014.

[49] Y. Tian, B. Liu, W. Dai, B. Ur, P. Tague, and L. F. Cranor. Supporting privacy-conscious app update decisions with user reviews. In *Security and Privacy in Smartphones and Mobile Devices*, 2015.

[50] J. Y. Tsai, S. Egelman, L. Cranor, and A. Acquisti. The effect of online privacy information on purchasing behavior: An experimental study. *Information Systems Research*, 22(2), 2011.

[51] Y. Wang, J. Zheng, C. Sun, and S. Mukkamala. Quantitative security risk assessment of Android permissions and applications. In *Data and Applications Security and Privacy XXVII*, 2013.

[52] Z. Wei and D. Lie. LazyTainter: Memory-efficient taint tracking in managed runtimes. In *Security and Privacy in Smartphones and Mobile Devices*, 2014.

[53] C. Welch. Tinder's new 'Social' feature reveals which Facebook friends are swiping - the Verge. Written Apr. 2016. `https://www.theverge.com/2016/4/27/11518034/tinder-social-reveals-swiping-facebook-friends`. Accessed: Apr. 2017.

[54] P. Wijesekera, A. Baokar, A. Hosseini, S. Egelman, D. Wagner, and K. Beznosov. Android permissions remystified: A field study on contextual integrity. In *USENIX Security Symposium*, 2015.

[55] L. Yang, N. Boushehrinejadmoradi, P. Roy, V. Ganapathy, and L. Iftode. Enhancing users' comprehension of Android permissions. In *Security and Privacy in Smartphones and Mobile Devices*, 2012.

[56] L. Yu, T. Zhang, X. Luo, and L. Xue. AutoPPG: Towards automatic generation of privacy policy for Android applications. In *Security and Privacy in Smartphones and Mobile Devices*, 2015.

[57] W. Zhou, Y. Zhou, X. Jiang, and P. Ning. Detecting repackaged smartphone applications in third-party Android marketplaces. In *Conference on Data and Application Security and Privacy*, 2012.

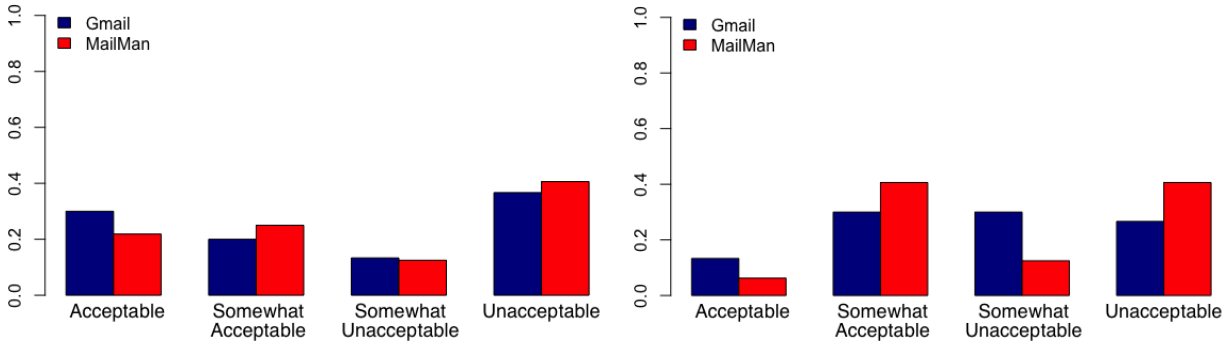
[58] Y. Zhou, Z. Wang, W. Zhou, and X. Jiang. Hey, you, get off of my market: Detecting malicious apps in official and alternative android markets. In *NDSS*, 2012.

Chapter 10

Appendices

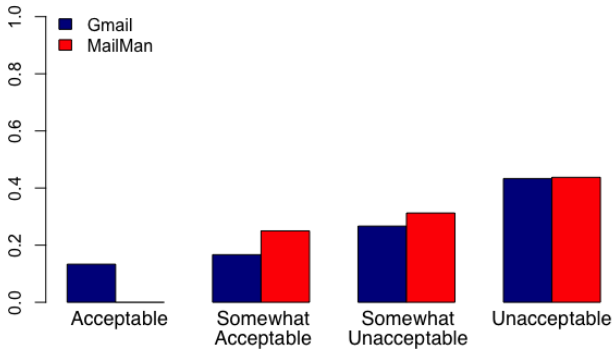
game	**business**	**medical**
Blossom Blast Saga	Job Search	CareZone
Star Wars: Galaxy of Heroes	ADP Mobile Solutions	MyChart
Clash of Kings	UPS Mobile	FollowMyHealth Mobile
Prize Claw 2	LinkedIn Job Search	Ovia Pregnancy Tracker
Subway Surfers	Job Search - Snagajob	ScriptSave WellRx
entertainment	**health_and_fitness**	**finance**
Netflix	Strava Running and Cycling GPS	Credit Karma
Hulu	Calorie Counter - MyFitnessPal	Chase Mobile
Google Play Games	CVS/pharmacy	Bank of America
Vine - video entertainment	Google Fit - Fitness Tracking	Android Pay
YouTube Kids	Headspace - meditation	PayPal
news_and_magazines	**social**	**music_and_audio**
Yahoo - News, Sports & More	Facebook	Pandora Radio
CNN Breaking US & World News	Instagram	Spotify Music
Viewers to Volunteers	Snapchat	SoundCloud - Music & Audio
AOL: Mail, News & Video	Pinterest	YouTube Music
Fox News	Twitter	Shazam
travel_and_local	**weather**	**white noise***
Waze - GPS, Maps & Traffic	The Weather Channel	White Noise Free
Yelp	1Weather:Widget Forecast Radar	White Noise Pro 2.0
Maps	AccuWeather	White Noise Baby
United Airlines	Transparent clock & weather	Relax Melodies: Sleep & Yoga
Southwest Airlines	WeatherBug	Relax Rain - Nature sounds
brick-and-mortar*		
Stop and Shop		
HSBC		
Wegmans		
Starbucks		
Subway		
Regal Cinemas		

Figure 10.1: Apps considered in the classification study (Chapter 3). Categories marked by an asterisk are not built-in Google Play categories but rather sets of apps with specific qualities of interest to the study: The "white noise" apps have very similar feature sets, and therefore might be likely to be considered generic by users, while apps in the "brick-and-mortar" category are closely coupled with real-world products and so might be likely to be single-source. ("Brick-and-mortar" is not mutually exclusive with respect to the other categories, so there are some apps in other categories that are "brick-and-mortar," such as CVS/pharmacy in `health_and_fitness` and the airline apps in `travel_and_local`.)

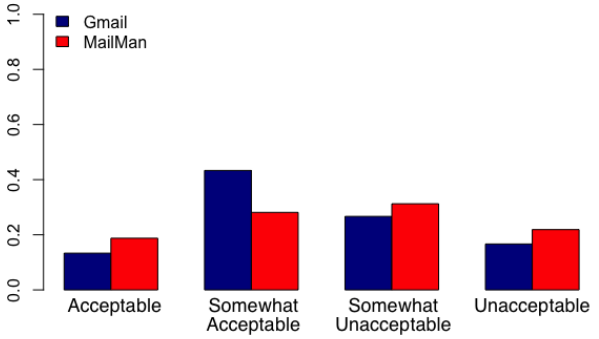


(a) The `Calendar` permission group.

(b) The `Photos/Media/Files` permission group.

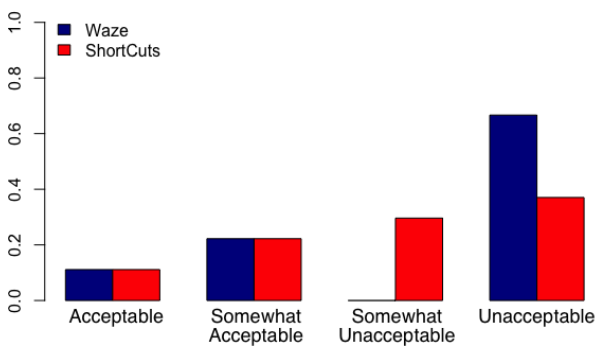


(c) The `Other` permission group.

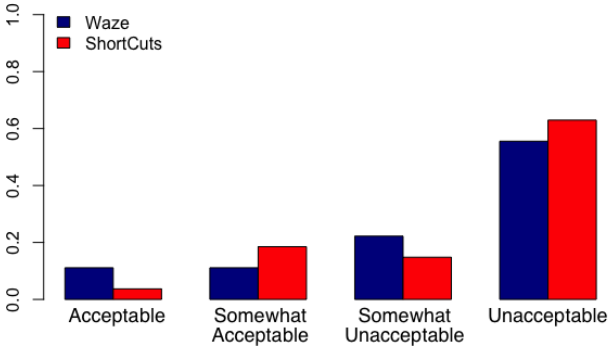


(d) The `Identity` permission group.

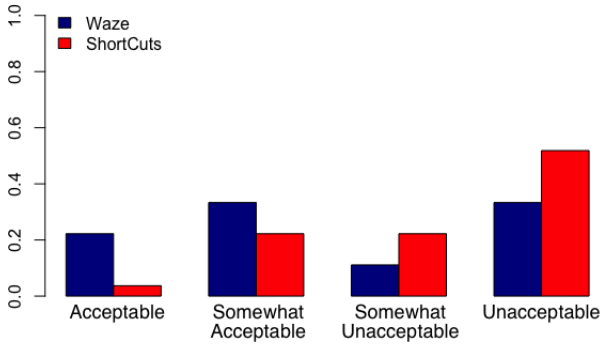
Figure 10.2: The ratings for each permission for the Gmail and MailMan apps.



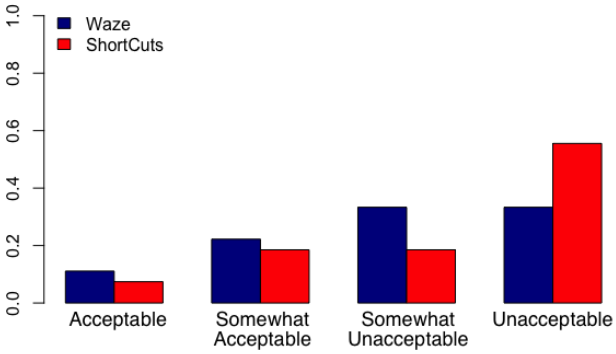
(a) The `Calendar` permission group.



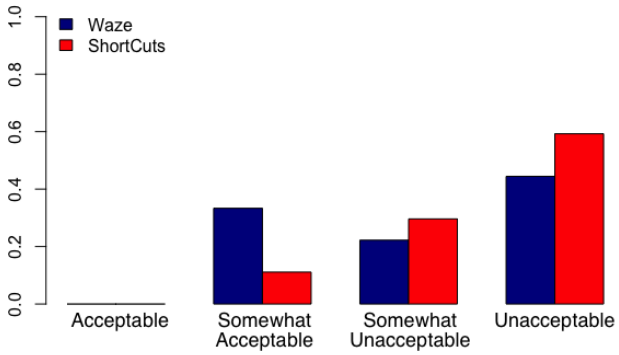
(b) The `Photos/Media/Files` permission group.



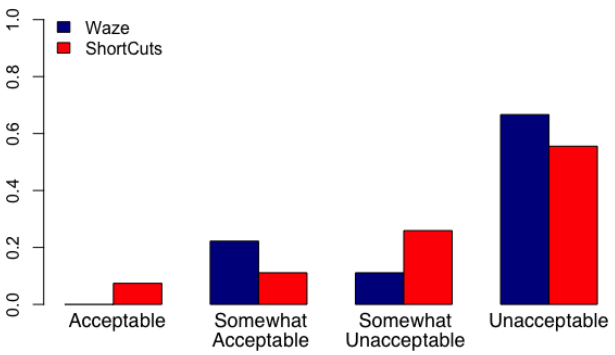
(c) The `Other` permission group.



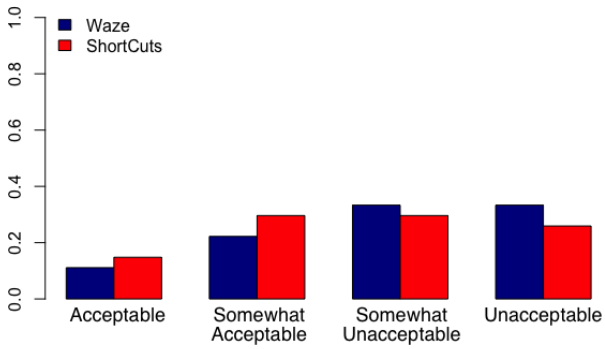
(d) The `Identity` permission group.



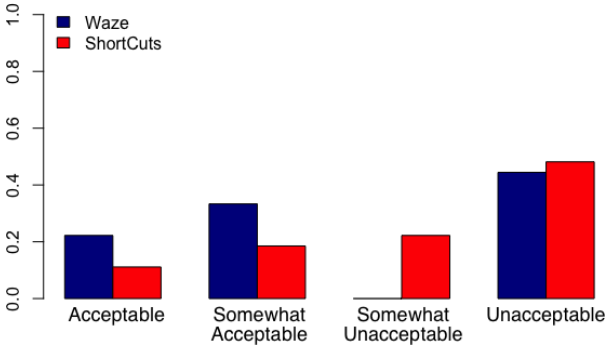
(e) The `Contacts` permission group.



(f) The `SMS` permission group.

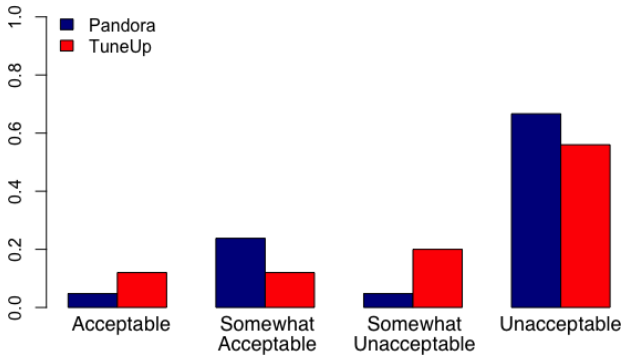


(g) The `Phone` permission group.

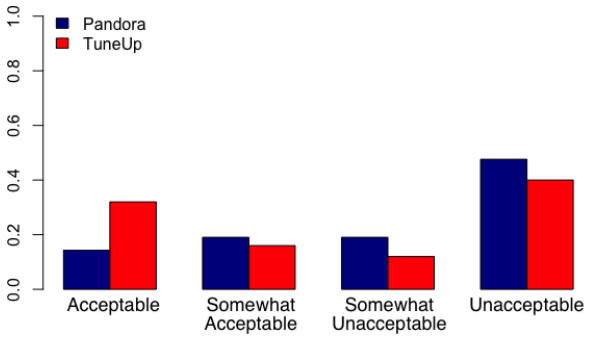


(h) The `Camera/Microphone` permission group.

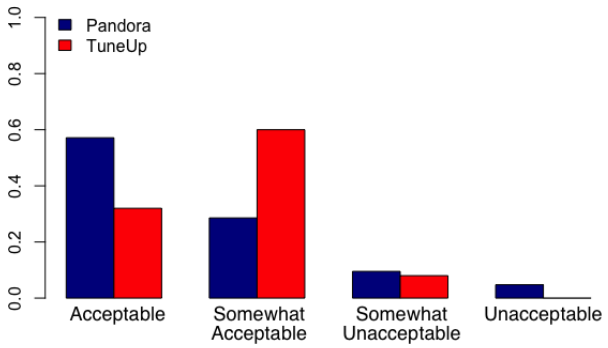
Figure 10.3: The ratings for each permission for the Waze and ShortCuts apps. After eliminating participants who had not heard of Waze, a brand-name app, the Waze condition had only 9 participants.



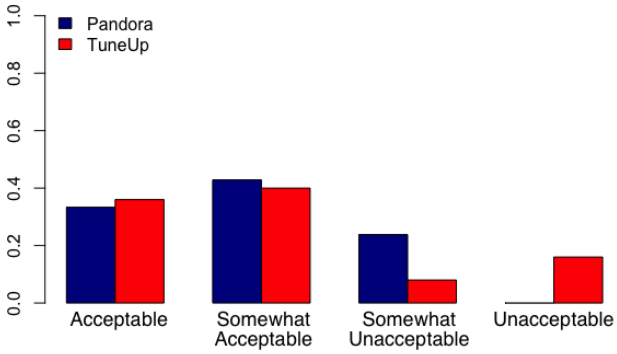
(a) The `Calendar` permission group.



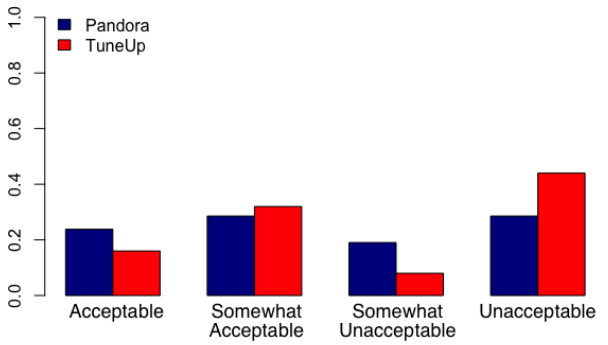
(b) The `Photos/Media/Files` permission group.



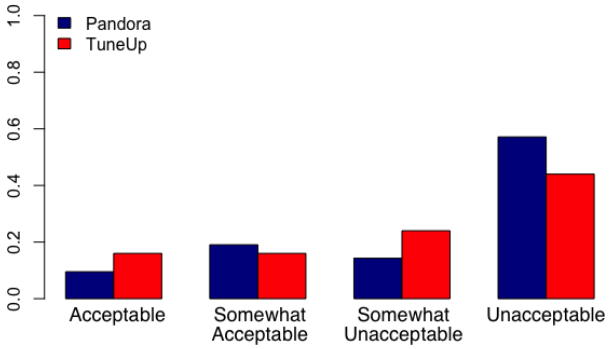
(c) The `Other` permission group.



(d) The `In-app purchases` permission group.

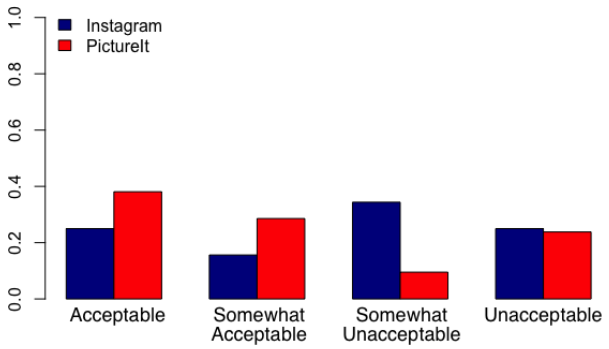


(e) The `Phone` permission group.

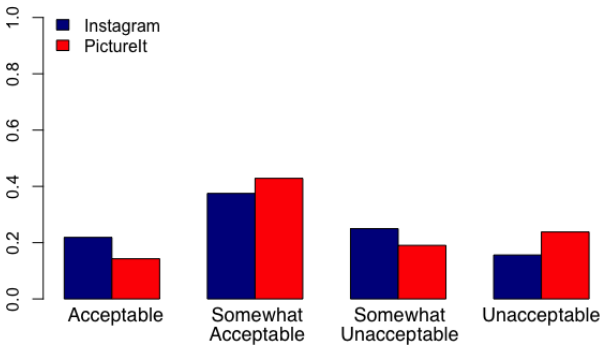


(f) The `Contacts` permission group.

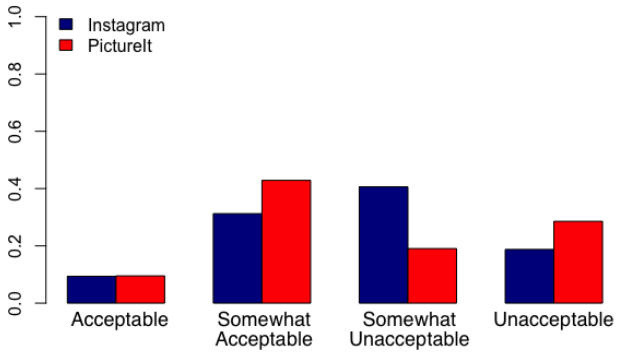
Figure 10.4: The ratings for each permission for the Pandora and TuneUp apps.



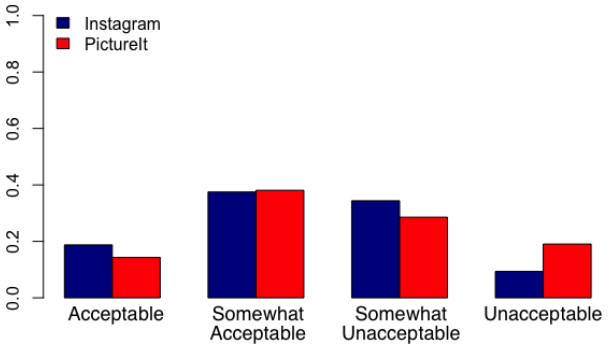
(a) The `Photos/Media/Files` permission group.



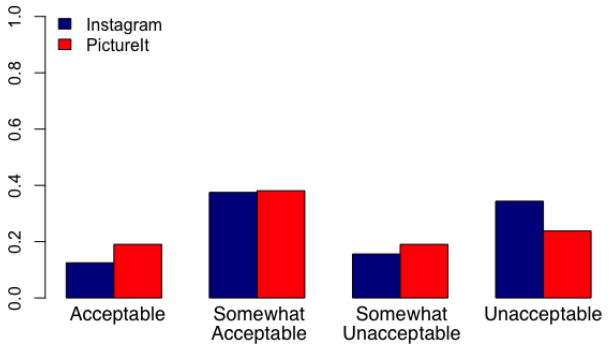
(b) The `Other` permission group.



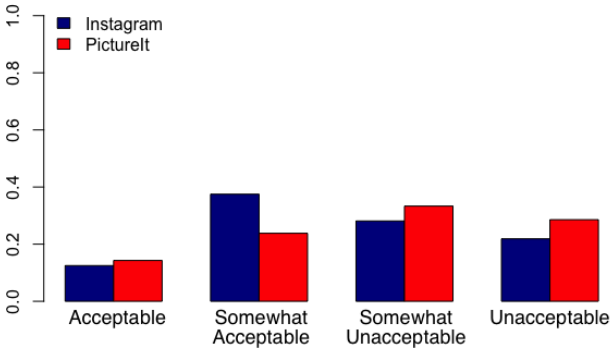
(c) The `Device & app history` permission group.



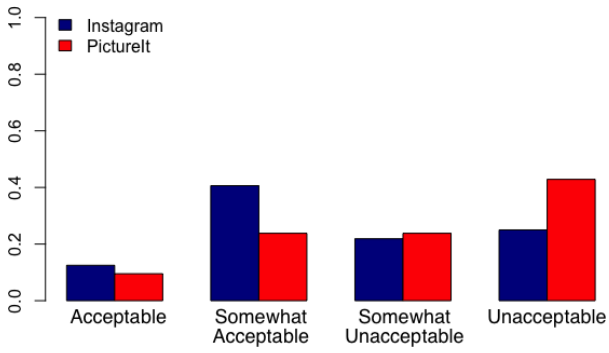
(d) The `Location` permission group.



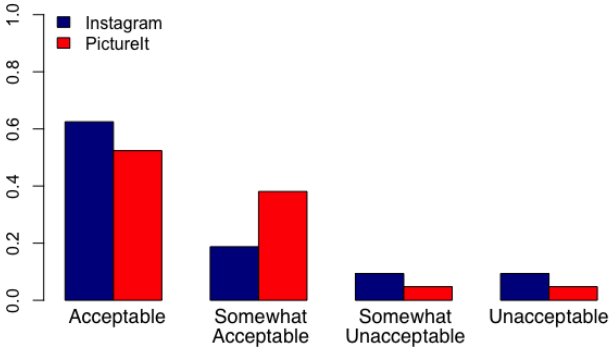
(e) The `Contacts` permission group.



(f) The `SMS` permission group.



(g) The `Phone` permission group.



(h) The `Camera/Microphone` permission group.

Figure 10.5: The ratings for each permission for the Instagram and PictureIt apps.

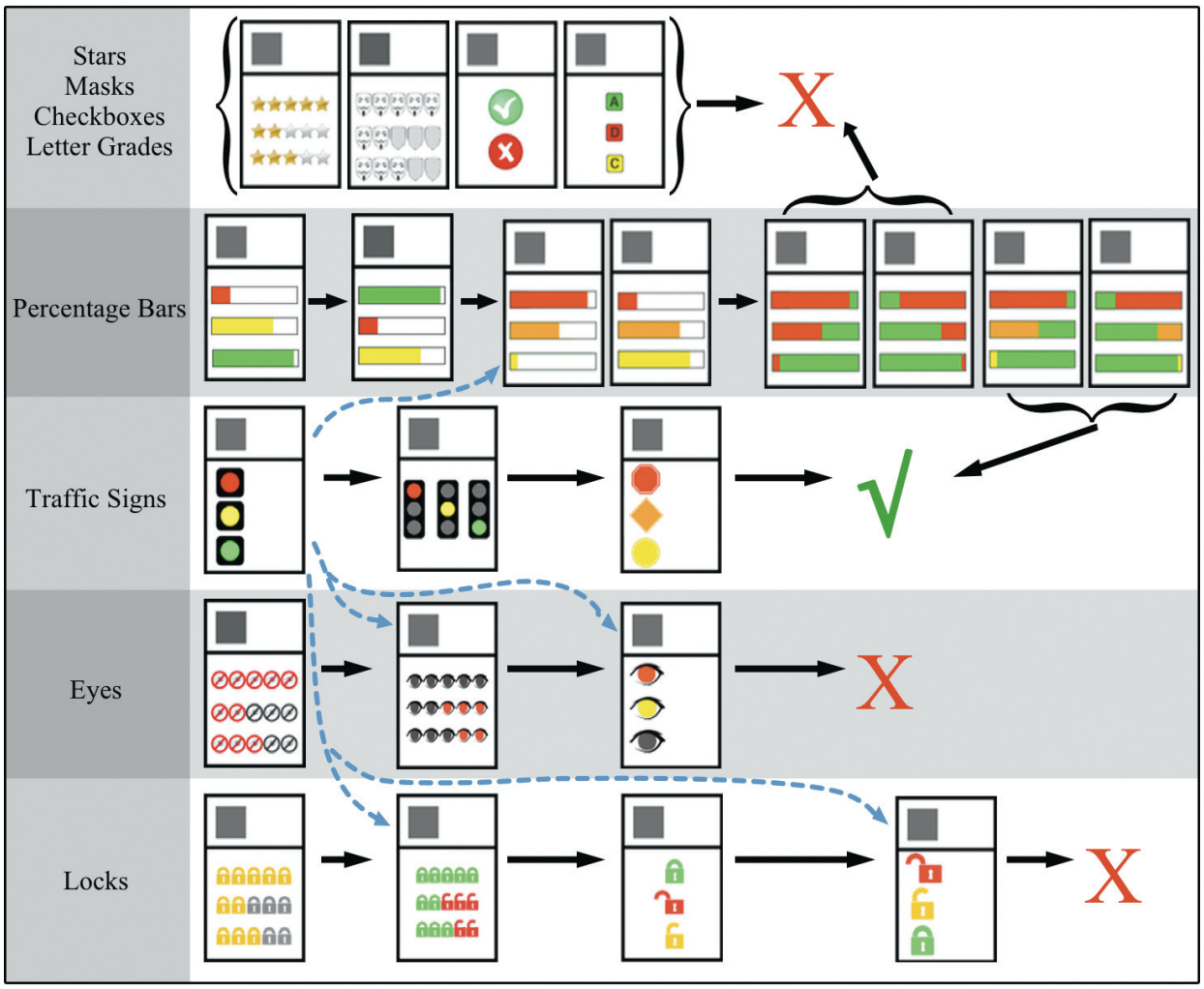


Figure 10.6: (Note: This figure may be better viewed in color.) An overview of all of the interfaces explored during our iterative design process (Chapter 8). Arrows map the evolution and cross-influences of interfaces; solid (black) arrows show redesigns, and dashed (blue) arrows indicate that feedback on one iconography influenced the design of another. X's (in red) indicate the elimination of an iconography, while the checkmark (in green) signifies the interface was included in our in-depth testing.