PriWe: Recommendation for Privacy Settings of Mobile Apps based on Crowdsourced Users' Expectations

Rui Liu, Jiannong Cao, Lei Yang
Department of Computing
The Hong Kong Polytechnic Univerisity
Hong Kong
{*csrliu, csjcao, csleiyang*}*@comp.polyu.edu.hk*

Kehuan Zhang
Department of Information Engineering
The Chinese University of Hong Kong
Hong Kong
khzhang@ie.cuhk.edu.hk

***Abstract*—Privacy is a pivotal issue of mobile apps because there is a plethora of personal and sensitive information in smartphones. Various mechanisms and tools are proposed to detect and mitigate privacy leaks. However, they rarely consider users' preferences and expectations. Users hold various expectations towards different mobile apps. For example, users can allow a social app to access their photos rather than a game app because it is beyond users' expectation when an entertainment app gets the personal photos. Therefore, we believe it is vital to understand users' privacy expectations to various mobile apps and help them to mitigate privacy risks in the smartphone accordingly. To achieve this objective, we propose and implement PriWe, a system based on crowdsourcing driven by users who contribute privacy permission settings of their apps in smartphones. PriWe leverages the crowdsourced permission settings to understand users' privacy expectation and provides app specific recommendations to mitigate information leakage. We deployed PriWe in the real world for evaluation. According to the feedbacks of 78 users from the real world and 382 participants from Amazon Mechanical Turk, PriWe can make proper recommendations which can meet participants' privacy expectation and are mostly accepted by users, thereby help them to mitigate privacy disclosure in smartphones.**

***Keywords*-mobile privacy; crowdsourcing; recommendation**

I. Introduction

When you are installing a mobile app, will you give a second thought in front of a long permission list of data usage? Most people just touch the accept button to approve of the permissions without scrutinizing them [1]. As a consequence, abuse of permission is common in many smartphone apps. For instance, some game apps require a plethora of data, including users' accounts, location information, personal photos and device ID, which may be not necessary for functioning. To most users, the expectation of a game app is entertainment rather than accessing so many kinds of information. Obviously, apps' behaviors may be beyond the users' expectations due to the abuse of the permissions. The risk of sensitive data disclosure is increased accordingly. To address this issue, it is significant to understand users' expectation of privacy and mitigate the abuse of data access permissions of mobile apps.

This problematic issue has got attentions from some giants of the smartphone world. They ameliorated the user interface of their systems to show some detailed information about data access permissions. The idea behind it is to improve users' understandings of the permissions rather than leaving the users in the dark. Unfortunately, most people think the information about data access permissions is intricate and few people will read them [2, 3]. Thus, a number of research projects have looked into understanding users' expectations of different mobile apps [4]. However, they captured the expectation and formed a unified conclusion without considering diversity of individual privacy expectation. For example, when the majority of users believe an app should not access a specific data, this result will be regarded as a common conclusion for this app and delivered to all the users. However, a common conclusion agreed by the majority of users by no means represents that all the users should follow it. Therefore, it is important and challenging to understand individual expectation. To achieve this objective, we propose PriWe, a system which aims to understand users' expectations and help them to make proper decisions about the privacy settings of their smartphone apps accordingly. The basic idea is that PriWe collects users' privacy settings to various apps and finds the users who have similar privacy preferences. Then, PriWe makes recommendations to a user about permission settings based on the expectations of the users, who hold the similar privacy preferences.

In this paper, we collect users' settings of data access permissions by leveraging crowdsourcing. After collecting, how to understand users' expectation of privacy and help them to make decisions is another challenging issue. Inspired by the recommendation algorithm, we calculate similarity of different users and similarity of different apps. We consider the users who have high similarities will share analogous preferences. Likewise, users will hold similar preferences to some similar smartphone apps. According to this feature, PriWe can make app specific recommendations to users.

We implemented PriWe and deployed it in the real world. The implementation in this paper focuses on Android operating system since it holds the largest smartphone market

978-1-4673-7284-8/15 $31.00 © 2015 IEEE
DOI 10.1109/MS.2015.30

share. PriWe allows users to set their data access permissions to various Android apps and generates recommendations based on their expectations by leveraging the crowdsourcing. We released the app of PriWe to 78 users and published a crowdsourcing task on the Amazon Mechanical Turk, which is completed by 382 participants. According to the results, PriWe can make proper recommendations which can meet participants' privacy expectations and are mostly accepted by users, thereby help them to make informed decisions about settings of data access permissions.

We make the following contributions in this paper: (1) We propose PriWe to understand users' expectation of privacy on Android apps using the crowdsourcing mechanism. (2) We implement and deploy PriWe in the real world. It allows users to set data access permissions of installed apps and apply the recommendations produced by PriWe to mitigate privacy leaks. (3) We evaluate PriWe by collecting the feedbacks of 78 users from the real world and 382 participants from the Amazon Mechanical Turk. According to the results, PriWe can make recommendations which are mostly accepted by users, thereby help them to make informed decisions and mitigate privacy disclosure.

II. Users' Expectation of Privacy

Taking a step back, we discuss the privacy in this section and figure out why understanding the individual expectation of privacy towards mobile apps is vital and beneficial.

Privacy is by no means a fad of modern society. In 1890, two U.S. lawyers proposed a prevalent definition, private life, habits, act, relations and the right to be alone [5]. With the proliferation of information technology, Wesin proposed that privacy is the claim of individuals, groups, or institutions to determine for themselves when, how, and to what extent information about them is communicated to others, and it came to be known as information privacy [6]. These two acknowledged definitions both emphasized that privacy to people should be an ability to express themselves selectively. The expression is driven by the individual expectation of privacy. Likewise, users also want to provide their data selectively based on their expectation to the privacy of mobile apps. On one hand, we yearn for better services and performances so that we are willing to provide some information. On the other hand, it is reluctant for us to share information because we also hope that our sensitive data could be preserved. Therefore, based on the previous discussion, it is a trade-off between service and privacy. People's expectation of privacy is the key to balance the trade-off.

III. System Design

In this section, we show the architecture of PriWe and elaborate on the mechanism we proposed to generate recommendations for privacy settings in smartphones.

A. Architecture

We have two intentions in our mind when designing PriWe. First, PriWe can be deployed in the real world and help users to make better decisions on privacy settings in their smartphones. Second, the processes of analyzing crowdsourced and generating recommendations should be completed in a server due to the limited capability of smartphones. To achieve these objectives, we design the system, as depicted in Fig. 1. A mobile app is deployed in the smartphone to collect privacy settings from users. It also can receive recommendations from the server to help user mitigate privacy risks. The analysis component in a server aims to analyze the crowdsourced data and generate the recommendations. All the data will be saved in an inbuilt database.

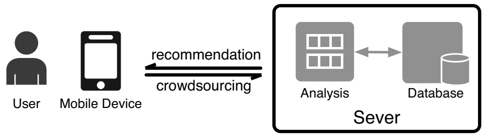


Figure 1. The overview of PriWe, which consists of an mobile app in the smartphone and a server.

The mobile app of PriWe should consist of several features. Firstly, it can automatically scan the apps installed in the smartphone and identify them by names. Secondly, the PriWe app can receive and apply the recommendations based on the crowdsourced information. Finally, the PriWe app itself should hold the data access permissions as little as possible. Because the ultimate objective of our project is to help users to mitigate privacy risk accordingly, our system should be a privacy problem in no event.

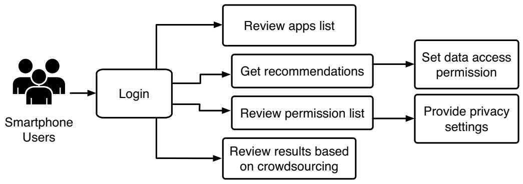


Figure 2. The main functions of the Android app of PriWe.

The main functions of PriWe app are shown in Fig. 2. After installing the app, a user needs to sign up a new account in order to use it. The user can then review the list of apps installed in the smartphone and the corresponding privacy permissions. Based on the crowdsourced other users' settings, the user can see the crowdsourcing results, receive and apply the recommendations generate by the server.

The server side of PriWe has two parts as shown in Fig. 1, one is a program run in the background to analyze the data and generate recommendations based on the crowdsourcing mechanism, the other one is an inbuilt database to store information. The analysis part plays a pivotal role on the server side and we elaborate on it subsequently.

B. Recommendation mechanism

For analyzing the users' crowdsourced data, we are inspired by recommendation algorithms. The traditional recommendation systems aim to recommend attractive and interesting commodities to customers in some e-commerce markets. However, we do not have customers and commodities; rather we have smartphone users and privacy settings. Thus, each user is mapped to a customer, and each privacy setting is mapped to a commodity. Furthermore, we consider that the people with the similar background, habit or age may have similar privacy preferences. In this section, we elaborate on generating recommendation by formalization in the context of smartphone apps.

We assume that there are P users, each user has A apps. Each app holds Q data access permissions. We define $I_{p,q,a}$ as the setting of data permission q of the app a set by the user p. The users set the privacy setting by the dichotomous variable $\{0,\ 1\}$. More specifically, $I_{p,q,a} = 0$ denotes that the users do not want share that data with anyone, whereas $I_{p,q,a} = 1$ means the participant allows the disclosure of that information. However, the users may not have a clear understanding to various privacy settings, and it is arduous for them to finish all of the privacy settings. The recommendations made by PriWe can help the users to set the data access permission if they have ambiguous understanding of the corresponding data. We propose the recommendation mechanism by drawing inspiration from the collaborative filtering approach, as portrayed in Fig. 3. More specifically, we adopted the user-based and item-based collaborative filtering. The following two examples further illustrate these two approaches.

Example 1: Two users, i and j, both installed two apps a_1, a_2 in the smartphone, and each app holds two permissions q_1, q_2. The user i and j both allow the app a_1 can get the corresponding data permissions, by setting $I_{i,q_{1,2},a_1} = 1$ and $I_{j,q_{1,2},a_1} = 1$. In this situation, we consider they may have the similar privacy preferences. If the user i set $I_{i,q_1,a_2} = 0$ to prohibit the access permission q_1 of the app a_2, user j is likely to have the same choice on this setting.

Example 2: Two apps, a_1 and a_2, both are installed in the smartphone carried by a user i'. The apps i and j have the similar functions and hold the same permissions q_1 and q_2. If the user set $I_{i',q_1,a_1} = 1$ to allow the corresponding data to be accessed by the app a_1, the user is very likely to do the same thing to the app a_2,by setting $I_{i',q_1,a_2} = 1$, as well.

The examples illustrate the user-based and item-based collaborative filtering approaches we applied. Based on the examples. the key of the two approaches is to calculate the similarity of users and items, respectively, and then generate recommendations. Therefore, we define $s_u(i,j)$ as the similarity between the user i and j. The basic idea of the similarity is that we want to discover how many privacy settings that the users i and j have the same choice. The more such settings, the higher similarity between them. Thus, we calculate similarity between user i and j through Eq. 1, based on the Pearson correlation coefficient. The possible similarity values are between -1 and 1, where values near to 1 indicate a strong similarity. The l served as a penalty parameter where n_k is defined as the number of users who have mark data usage permission k as private and n is total number of users. If everyone has marked permission k, the l is zero. The l can avoid the influence if the users make general consensus on permission k. Furthermore, to improve the accuracy of results, we also consider various basic information of the users, including, occupation, age, gender and smartphone daily usage. More specifically, we thought that the users have the similar basic information may have the similar privacy preferences. More specifically, the PriWe will first categorized the users into different groups due to their background, habit and age. Then, the PriWe calculate the similarity between the users in the same group and those in the different group afterwards.

$$s_u(i,j) = \frac{\sum\limits_{k\in Q} l \sum\limits_{k\in Q} lI_{i,k,a}I_{j,k,a} - (\sum\limits_{k\in Q} lI_{i,k,a})(\sum\limits_{k\in Q} lI_{j,k,a})}{\sqrt{MN}} \tag{1}$$

Where

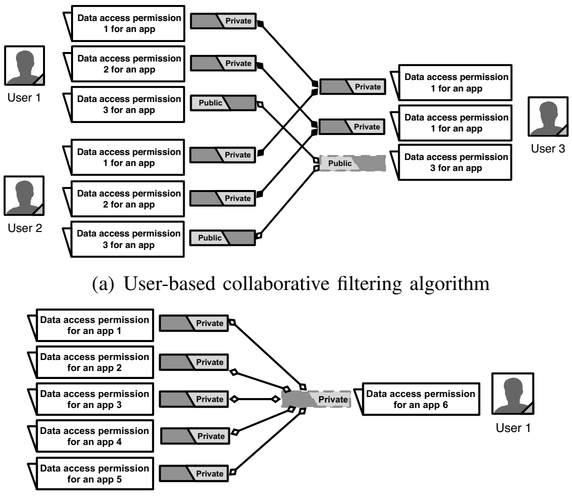
$$\begin{aligned} M &= \sum_{k\in Q} l(\sum_{k\in Q} lI_{i,k,a}^2 - (\sum_{k\in P} lI_{i,k,a})^2) \\ N &= \sum_{k\in Q} l(\sum_{k\in Q} lI_{j,k,a}^2 - (\sum_{k\in P} lI_{j,k,a})^2) \\ l &= log\frac{n}{n_k} \end{aligned}$$

Likewise, we define $s_i(u,v)$ as the similarity between the privacy permission u and v. To calculate the similarity, we adopted the adjusted cosine similarity to take the differences in the average setting behaviors of the users into account, as shown in Eq. 2.

$$s_i(u,v) = \frac{\sum\limits_{i\in P}(I_{i,u,a} - \overline{I}_{u,a})(I_{i,v,a} - \overline{I}_{v,a})}{\sqrt{\sum\limits_{i\in P}(I_{i,u,a} - \overline{I}_{u,a})^2}\sqrt{\sum\limits_{i\in P}(I_{i,v,a} - \overline{I}_{v,a})^2}} \tag{2}$$

The $\overline{I}_{u,a}$ denotes the average of permission setting u of the app a set by all the users. The results for the adjusted cosine measure correspondingly range from 1 to +1, as in the Pearson measure. We adopted Pearson correlation coefficient and the adjusted cosine similarity to calculate the similarity between users and permission settings, respectively. The empirical analysis showed that for user-based recommender systems by far, the Pearson correlation coefficient outperforms other measures of comparing users [7]. However, it has been presented that the adjusted cosine similarity consistently outperforms the Pearson correlation metric in the item-based recommendation situations.

After calculating the similarities $s_u(i, j)$ and $s_i(u, v)$, we adopt a probabilistic-based similarity fusion framework [8] to form a more robust similarity and overcome the data sparsity problem, which is an obstacle to our work for real-world deployment. The basic idea is that we provide different weights to the two similarities $s_u(i, j)$ and $s_i(u, v)$ based on probability, and combine them accordingly. Thus, the user-based and item-based collaborative filtering approaches are only two special cases. Finally, we can produce the recommendation to each user for their privacy permission due to the similarities.


Data access permission 1 for an app
Data access permission 2 for an app
Data access permission 3 for an app
User 1
Data access permission 1 for an app
Data access permission 2 for an app
Data access permission 3 for an app
User 2
Private
Private
Public
Private
Private
Public
Private
Private
Public
Data access permission 1 for an app
Data access permission 1 for an app
Data access permission 3 for an app
User 3
Data access permission for an app 1
Data access permission for an app 2
Data access permission for an app 3
Data access permission for an app 4
Data access permission for an app 5
Private
Private
Private
Private
Private
Private
Data access permission for an app 6
User 1

(a) User-based collaborative filtering algorithm

(b) Item-based collaborative filtering algorithm

Figure 3. Generating recommendation of data access permissions for Android apps is based on the user- and item-based collaborative filtering algorithm.

Table I
SUMMARY OF MOST ABUSED DATA AND PERMISSIONS

Most Abused Data and Permissions
• **Coarse and fine location** (Approximate or exact location information. It can lead location-based attacks or malware, or sending location-based ads.)
• **Network state** (Cellular network information and connections. It will also drain smartphones' battery.)
• **Wifi network information** (Wi-Fi network information, including passwords and usernames. It can lead information disclosure by Wi-Fi network.)
• **Running apps information** (Information of running tasks and processes. Users' sensitive information from other running apps can be leak.)
• **Phone state and identity** (Phone states information and International Mobile Equipment Identity. It can lead sensitive information disclosure.)
• **Modify/Delete internal/external contents** (Permission of modification internal and external storage. Apps steal information or save data on internal and external storage.)
• **Full internet access** (Permission of using the Internet to download and upload. The sensitive information can be disclosed and malware will be downloaded.)
• **Automatically Start at Boot** (Permission of automatically starting the smartphones boot. Malicious apps will use it to boot automatically.)
• **Send SMS Messages** (Permission of sending text messages without users' awareness for subscribe additional services which may leave users with unexpected charges.)
• **Prevent From Sleeping** (Permission of preventing from sleeping or the screen from dimming. Apps can steal the information even it is time-consuming.)
• **Control Vibrator** (Permission of accessing vibrator function. It can stop vibrations for notification before malicious apps interpret information.)

IV. IMPLEMENTATION

In this section, we describe the implementation of PriWe, from the app side and server side.

A. *PriWe App*

The PriWe app is developed in Android platform and can be compatible with various smartphones with different screen size. Our prototype of PriWe app is implemented on Android 4.4 and runs on the Google/LG Nexus 4 handset. For deployment, PriWe has been tested with Android system version 4.0.3 - 4.4.4.

The two main objectives of PriWe app are to provide user-interface for setting their apps' permissions of data access, and to collect users' settings based on the crowdsourcing. More specifically, the main functions of PriWe app includes signing in, setting apps' permissions, collecting users' feedbacks, applying recommendations and presenting the results. The functions of the app are depicted in Fig. 4, which is composed by snapshots of the app in Nexus 4. All functions and interactions between them are implemented by Activity and Fragment. The official Android SDK provides APIs to retrieve installed apps in the smartphone and scrutinize the privcacy permissions of each app. However, it is arduous for users to read all the system permissions in a screen of smartphone. we summarize eleven types of abused data and permissions of Android apps and discuss their potential risks, as shown in Table I. The summary is based on some freeform comments on the Internet [9, 10], research papers on the Android system and analysis of smartphone apps [1, 2, 3, 11, 12], security and privacy tips from official guidelines [13], a survey on information security and privacy of Android apps [14].

To implement the function of setting permission in Android platform, PriWe provides user interface to set permissions. The Android system does not provide any mechanism for users to modify the privacy settings to normal users. PriWe can achieve such functions based on the Xposed framework [15]. It allows the users to change the privacy permission settings for various mobile apps. Since apps may be crashed when they cannot access specific data, PriWe feeds Android apps artificial data. However, there are two exceptions: access to the internet and modify external storage are restricted by denying access due to Android system mechanism. Overall, PriWe has the capability to modify the data access permission of installed apps in

Android smartphone. Thus, PriWe requires root permission, namely, there is no way to achieve our objectives in non-root devices. Although root process is considered as legal, it is not supported officially. We take this issue in a neutral way and we do not advocate rooting Android smartphone for protecting users' privacy. However, in our work, PriWe needs root permission to mitigate information disclosure. Furthermore, according to the feedbacks from users, we did find no users have reported issues about data leakage or system crashed due to rooting their smartphones.

B. *PriWe Server*

The server is designed to analyze the collected data and articulate the results according to crowdsourcing mechanism. The server system is deployed in an IBM server and built as three-tier architecture which is composed of a presentation tier, a domain logic tier, and a data persistence tier. More specifically, the presentation tier, is a web-front which implemented by Html, Javascript and the third development libraries. A user friendly interface can be provided in this tier. The domain logic tier is implemented by Java EE architecture and Enterprise Beans mechanism to analyze the collected data. To improve robustness and configurability of the system, the web application is built based on frameworks including Spring, Struts, Hibernate. The recommendation algorithm we presented before is also deployed in this tier to generate recommendations to the users. In data persistence tier, all data are persisted in a MySql database.

V. Evaluation

To evaluate PriWe, we published a task to collect people's feedbacks about the privacy of smartphones on the Amazon Mechanical Turk. Furthermore, to make the evaluation results more convincing, we also deployed the PriWe in the real world. We elaborate on the evaluation from these two parts, respectively.

A. *Evaluation based on mechanical turk*

We published a task on the Amazon Mechanical Turk[1] for three weeks, and 382 participants completed our task. To avoid bias and make the results more convincing, we present the statistics of the participants. Among all the participants, 243 participants are male, and 139 participants are female. 226 participants are 20-29 years old, and 115 participants are 30-39 years old. The remainder of the participants are either 10-19 or above 40. All of the participants came from various backgrounds, such as, energy, materials, consumer staples, health care, financials, information technology and etc.

In the task, we asked the participants to answer a questionnaire to illustrate their privacy preferences for various types of mobile apps. We prepare two questionnaires, survey A and survey B. A participant will arbitrarily select one of them. According to the results, 200 participants completed the survey A and 182 finished the survey B. We evaluate the accuracy of recommendations produced by PriWe. More specifically, we separate the survey A into two parts, one is regarded as a train set, the other one is regarded as a test set. So does the survey B. Furthermore, to make the results more convincing, we also treated the survey A as a train set and the survey B as the test set. The selected results are presented in Fig. 5.

We have selected several populations of the participants, such as male, female, 20-24, 25-29, 30-39, with a background in information technology with a focus on privacy & security and with a background in information technology without a focus on privacy & security. The overall accuracy of the recommendations made by PriWe is about 78%. We made a comparison of the recommendation made by PriWe with the test data, presenting the results in Fig. 5. The accuracy of recommendation for the participants in privacy and security is higher than the remainder of all the selected participants (around 90%), because the users who have the background about the information privacy and security have a better understanding about the privacy permission settings in smartphones. Due to the same reason, the users who came from other areas have the lowest accuracy of recommendations. Another finding is that accuracy increased gradually in the users who are from 20 to 40 years old. One potential explanation is that some young people have no unambiguous perceptions about their privacy permission of their mobile apps.

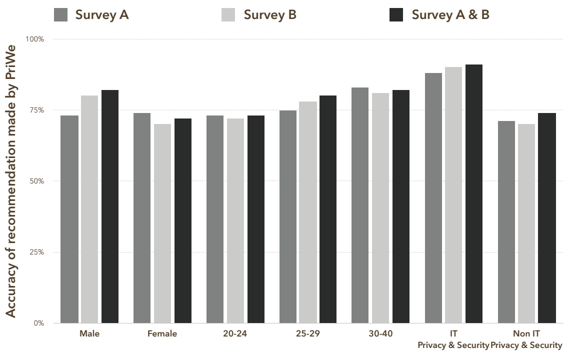
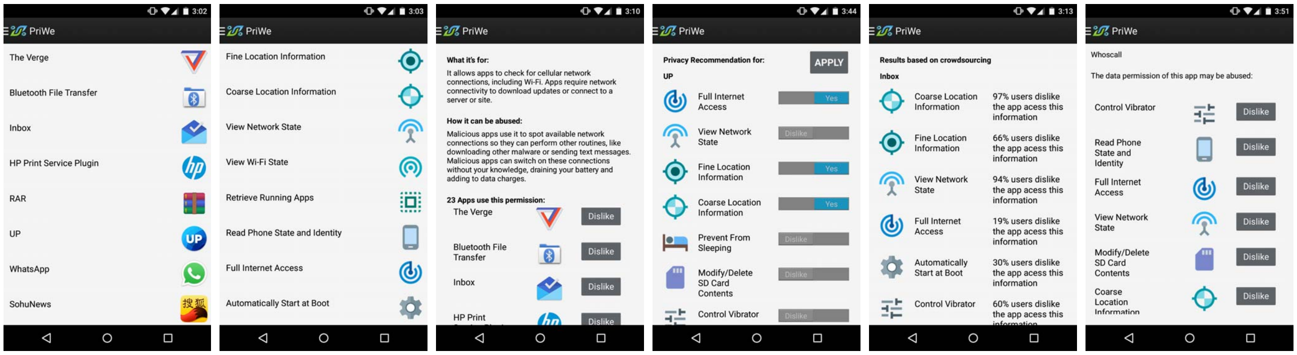


Figure 5. The accuracy of recommendation made by PriWe based on the feedbacks of participants in Amazon Mechanical Turk.

B. *Evaluation based on real-world deployment*

PriWe app has also been released to 78 users, who are from Hong Kong, Singapore, Austria, England, America and China, for evaluation in the real world. The server collected users' feedbacks of their permission settings and some basic information. The collected information includes app information, users' permission settings of installed apps, and the

[1] https://www.mturk.com/mturk/preview?groupId=3PBTVBPQ8T1PENG33V3IMPSHIB9LG1



The Verge
Bluetooth File Transfer
Inbox
HP Print Service Plugin
RAR
UP
WhatsApp
SohuNews

(a)

Fine Location Information
Coarse Location Information
View Network State
View Wi-Fi State
Retrieve Running Apps
Read Phone State and Identity
Full Internet Access
Automatically Start at Boot

(b)

What it's for:

It allows apps to check for cellular network connections, including Wi-Fi. Apps require network connectivity to download updates or connect to a server or site.

How it can be abused:

Malicious apps use it to spot available network connections so they can perform other routines, like downloading other malware or sending text messages. Malicious apps can switch on these connections without your knowledge, draining your battery and adding to data charges.

23 Apps use this permission:

The Verge — Dislike
Bluetooth File Transfer — Dislike
Inbox — Dislike
HP Print — Dislike

(c)

Privacy Recommendation for: APPLY

UP

Full Internet Access — Yes
View Network State — Dislike
Fine Location Information — Yes
Coarse Location Information — Yes
Prevent From Sleeping — Dislike
Modify/Delete SD Card Contents — Dislike
Control Vibrator — Dislike

(d)

Results based on crowdsourcing

Inbox

Coarse Location Information	97% users dislike the app acess this information
Fine Location Information	66% users dislike the app acess this information
View Network State	94% users dislike the app acess this information
Full Internet Access	19% users dislike the app acess this information
Automatically Start at Boot	30% users dislike the app acess this information
Control Vibrator	60% users dislike the app acess this information

(e)

Whoscall

The data permission of this app may be abused:

Control Vibrator — Dislike
Read Phone State and Identity — Dislike
Full Internet Access — Dislike
View Network State — Dislike
Modify/Delete SD Card Contents — Dislike
Coarse Location Information — Dislike

(f)

Figure 4. PriWe provides an Android app for participants. (a) PriWe can scan various app installed in smartphones; (b) PriWe also provides an user interface to the participants to list the most abused data access permissions; (c) The participants can discover how many installed apps used a specific permission and provide their privacy preferences; (d) The participants can also take a look about how many permissions an app will use and show their feedbacks of privacy preference accordingly; (e) The statistical results are presented to the participants, which can be taken as a reference for their privacy preferences; and (f) PriWe can make recommendations to various apps according to the individual privacy preferences.

users' basic information, such as background, age, gender, user ID and etc. In the evaluation, we collected information from 78 participants based on PriWe. The summary of apps of each user is shown in Table II. From the table, it can be seen that the majority of the users have less than 40 apps in their smartphone, which almost meets the statistic of common users' apps from Statistics Portal [16].

Table II
STATISTICS OF PARTICIPANTS' ANDROID APPS

Number of apps	Number of users	Percentage
1∼20	26	33%
20∼40	27	35%
40∼60	17	22%
60∼	8	10%

To corroborate the proposed abused data and permissions list, we calculate the average number of Android apps that participants installed access these data and permissions. According to the results presented in Table III, we found that all the potential abused privacy permissions have been accessed by many apps. Considering Table II and III jointly, we can figure out the majority of apps used by participants hold the abused permissions.

We illustrated the evaluation results in Fig. 6 and Fig. 7. So far there is no clear or existing metric to evaluate our work, we consider the participants' feedbacks as the ground truth to evaluate PriWe. According to Fig. 6, the recommendations are usually taken by the users. However, the recommendations about preventing from sleeping and controlling vibrator are not fully apprehended and reluctant to be applied by users. The reason of this phenomenon may be that they are not very severe risks and participants did not take much attention to them, ignoring the preferences and recommendations. The recommendations about location, network state and wifi network information, running apps and automatically starting are highly accepted. Participants may take them seriously since these information involved personal and even sensitive data. That is a reason why participants are willing to take them. Furthermore, participants showed ambivalence about the recommendations of phone state and identity, modify storage contents, Internet capability and SMS messages control. Because these information or permissions plays important roles in apps running and service performances, the ambivalence presents participants hope to obtain better services and preserved these information as well.

Table III
THE AVERAGE NUMBER OF ANDROID APPS THAT ACCESS ABUSED INFORAMTION

Abused data and permissions	Number of apps
Coarse and fine location	16
Network state	32
Wifi network information	20
Running apps information	13
Phone state and identity	18
Modify/Delete contents	30
Full internet access	35
Automatically start at boot	17
Send SMS messages	7
Prevent from sleeping	25
Control vibrator	27
Access 2∼5	27
Access 6∼10	16
Access all	5

To evaluate our another objective, i.e., improving awareness of privacy preference, we depicted the results according to the feedbacks in Fig. 7. From the graph, we can see that participants have a better comprehend or even epiphany to some privacy permissions. However, the participants did not have a better understanding about the permission of automatically boot and wifi network information. According to the survey after the experiment, we discovered that most participants already knew some mobile apps can boot automatically so they did not pay more attention to it. The

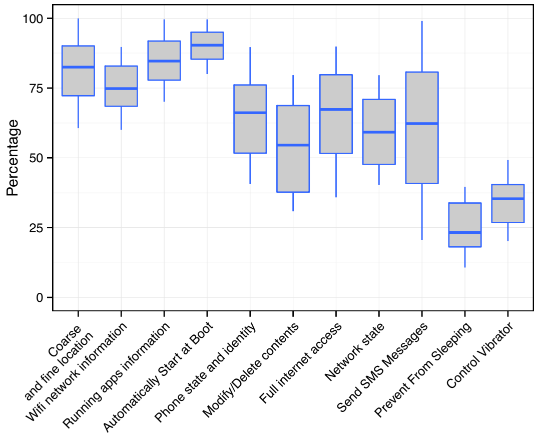


Figure 6. The percentage of apps that users take the recommendations of each data permission.

wifi network is permeating our life in every aspect inevitably and people take it as a kind of routine. Thus, participants did not feel a remarkable improvement of awareness of wifi network information.

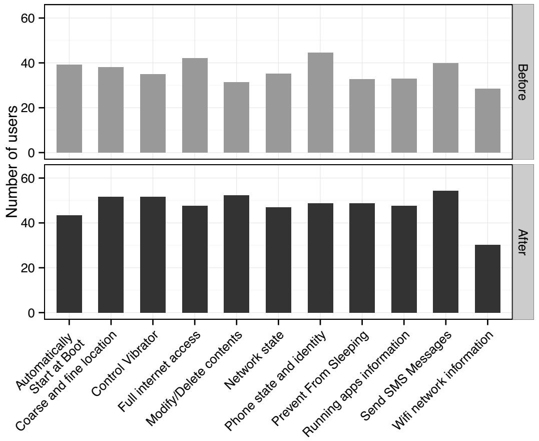


Figure 7. The number of users have a better understanding of each data access permission after using PriWe.

VI. Related Work

We provide an overview of some of representative literature related to our work in this section. We classify the related work into three categories: (1) security protection for Android apps (2) Android permission (3) understanding privacy and decision making system.

A. Security protection for Android apps

According to recent systematically research, several vulnerabilities have existed in Android apps. Their presence even in some extremely popular apps [17]. Thus, plenty of work focuses on security and privacy of Android platform and its apps. Techniques and tools that can detect and prevent information from being leaked in Android apps have been widely studied [18, 19, 20, 21]. Permission analysis is a telling method to detect sensitive information potential leakage [21]. Some static analysis tools have also been developed to automatically detect attempts to load external code using static analysis techniques [18]. Access control provides a different perspective of security and privacy detection and protection in Android system. FlaskDroid [19] privodes mandatory access control on Android's middleware and kernel layers to prevent information disclosure. TaintDroid [20] is a notable dynamic taint tacking and analysis system, which involves some aforementioned methods to simultaneously tracking multiple sources of sensitive data. All these works put much efforts on analyzing and protecting security for Android apps. However, protecting users' information unilaterally cannot meet their requirements since users have different concerns towards various mobile apps.

B. System permissions of Android apps

Android provides security to users through a permission mechanism [13]. The basic idea behind the permission mechanism is that each application has permissions to perform any operations that would adversely influence other applications, the system and users. The permission list of an app will be shown to users before they install apps from the app store. Only when the apps get approbation does they can be installed. The apps can access the information according to their permission lists.

Obviously, Android permission mechanism intends to improve users' awareness of the privacy about the apps. However, most Android users have defective understandings about the permission. To make things worse, they paid limited attentions to the permission list which is shown on the screen just before installation [1]. Thus, a mechanism, called permissions removal, has been proposed to mitigate the privacy leak in Android smartphone [3]. Another feasible way to mitigate data abuse is to establish a system with the ability to prevent apps from accessing resources without the stated permissions [11]. In this case, users will know what kind of information will be obtained by the app. However, some developers always ask for unnecessary permissions due to ambiguous API documentations and bad develop habits [12]. This abuse of permissions also lead unexpected information disclosure. Thus, static analysis of android permission can figure out the flaws when applications are granted more permissions than they actually need [22].

C. Understanding privacy and decision making system

According the discussion in section II, the privacy of Android apps should emphasize that users have adequate awareness and understanding to their personal, even sensitive information. According to a recent survey [23], Android users hold quite different viewpoints due to their demographic characteristics, security and privacy awareness, and reported behavior when installing apps. It is challenging to recognize users perceptions of whether a given action is legitimate, or how the action makes them feel with respect to privacy. A model, privacy as expectations [4], is proposed to capture people's expectations of privacy.

Past work about understanding privacy of smartphone users mostly take advantage of crowdsourcing. Our proposed system, PriWe, is inspired by these works. However, it differs in the motivation and the way of collecting and analyzing data. PriWe captures the information through an Android app and learn users' privacy concerns and preferences based on the collected data. Hence, PriWe makes recommendations to users based on their expectations.

VII. Conclusion

In this paper, we proposed PriWe, a system aims to understanding users' expectations of privacy and making recommendations about their privacy settings of installed mobile apps accordingly. We published a task on the Amazon Mechanical Turk and deployed PriWe in the real world for evaluation. According to the feedbacks of 382 participants from the Amazon Mechanical Turk, the recommendation made by PriWe can achieve around 78% accuracy for all the participants and achieve about 90% accuracy for the people in information privacy and security area. According to the feedbacks of 78 users from the real world, PriWe can make proper recommendations which can meet participants' privacy expectation and are mostly accepted by users, thereby help them to mitigate privacy disclosure in smartphone apps.

Acknowledgment

We thank the reviewers for their insightful comments. The work is supported by the NSFC/RGC Joint Research Scheme No. N_PolyU519/12 and the National Natural Science Foundation of China with Grant 61332004.

References

[1] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetherall, "A conundrum of permissions: installing applications on an android smartphone," in *Financial Cryptography and Data Security*. Springer, 2012, pp. 68–79.

[2] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proceedings of the Eighth Symposium on Usable Privacy and Security*. ACM, 2012, p. 3.

[3] Q. Do, B. Martini, and K.-K. R. Choo, "Enhancing user privacy on android mobile devices via permissions removal," in *System Sciences (HICSS), 2014 47th Hawaii International Conference on*. IEEE, 2014, pp. 5070–5079.

[4] 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 *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 501–510.

[5] S. D. Warren and L. D. Brandeis, "The right to privacy," *Harvard law review*, vol. 4, no. 5, pp. 193–220, 1890.

[6] A. F. Westin, "Privacy and freedom," *Washington and Lee Law Review*, vol. 25, no. 1, p. 166, 1968.

[7] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 1999, pp. 230–237.

[8] J. Wang, A. P. De Vries, and M. J. Reinders, "Unifying user-based and item-based collaborative filtering approaches by similarity fusion," in *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2006, pp. 501–508.

[9] "12 Most Abused Android App Permissions." http://about-threats.trendmicro.com/us/library/image-gallery/12-most-abused-android-app-permissions, 2013.

[10] "92% of top 500 android apps carry security or privacy risk." http://www.infosecurity-magazine.com/news/92-of-top-500-android-apps-carry-security-or/, 2014.

[11] C. Orthacker, P. Teufl, S. Kraxberger, G. Lackner, M. Gissing, A. Marsalek, J. Leibetseder, and O. Prevenhueber, "Android security permissions–can we trust them?" in *Security and Privacy in Mobile Information and Communication Systems*. Springer, 2012, pp. 40–51.

[12] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 627–638.

[13] "Android System Permissions." http://developer.android.com/guide/topics/security/permissions.html, 2014.

[14] X. Jiang and Y. Zhou, "A survey of android malware," in *Android Malware*. Springer, 2013, pp. 3–20.

[15] "Xposed Module Repository." http://repo.xposed.info/.

[16] "The Statistics Portal." https://www.statista.com/search/?statistics=1&studies=1&industryReports=1&dossiers=1&infos=1&subCategory=0&interval=0&category=0&subCategory=0&region=0&price=0&archive=0&q=how+many+apps+in+the+smartphone&sortMethod=idrelevance&accuracy=and&itemsPerPage=25, 2014.

[17] Y. Zhou and X. Jiang, "Detecting passive content leaks and pollution in android applications," in *NDSS*, 2013.

[18] S. Poeplau, Y. Fratantonio, A. Bianchi, C. Kruegel, and G. Vigna, "Execute this! analyzing unsafe and malicious dynamic code loading in android applications," in *NDSS*, vol. 14, 2014, pp. 23–26.

[19] S. Bugiel, S. Heuser, and A.-R. Sadeghi, "Flexible and fine-grained mandatory access control on android for diverse security and privacy policies," in *Usenix security*, 2013, pp. 131–146.

[20] 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 real-time privacy monitoring on smartphones," *Communications of the ACM*, vol. 57, no. 3, pp. 99–106, 2014.

[21] D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to android," in *Proceedings of the 17th ACM conference on Computer and communications security*. ACM, 2010, pp. 73–84.

[22] J. Klein, M. Monperrus, A. Bartel, and Y. Le Traon, "Static analysis for extracting permission checks of a large scale framework: The challenges and solutions for analyzing android," *IEEE Transactions on Software Engineering*, p. 1, 2014.

[23] Z. Benenson, F. Gassmann, and L. Reinfelder, "Android and ios users' differences concerning security and privacy," in *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 2013, pp. 817–822.