Understanding Mobile Users' Privacy Expectations: A Recommendation-Based Method Through Crowdsourcing

Rui Liu, *Member, IEEE*, Junbin Liang, Jiannong Cao, *Fellow, IEEE*, Kehuan Zhang, *Member, IEEE*, Wenyu Gao, Lei Yang, *Member, IEEE*, and Ruiyun Yu, *Member, IEEE*

Abstract—Privacy is a pivotal issue of mobile apps because there is a plethora of personal and sensitive information in smartphones. Many mechanisms and tools are proposed to detect and mitigate privacy leaks. However, they rarely consider users' preferences and expectations. Users hold various expectation towards different mobile apps. For example, users may allow a social app to access their photos rather than a game app because it goes beyond users' expectation to access personal photos. Therefore, we believe it is practical and beneficial to understand users' privacy expectations on various mobile apps and help them mitigate privacy risks introduced by smartphones. To achieve this objective, we propose and implement PriWe, a system based on crowdsourcing driven by users who contribute privacy permission settings of the apps installed on their smartphones. PriWe leverages the crowdsourced permission settings to understand users' privacy expectations and provides app specific recommendations to mitigate information leakage. We deployed PriWe in the real world for evaluation. According to the feedback of 78 users who evaluated our system and 422 participants who completed our survey, PriWe is able to make proper recommendations which can match participants' privacy expectations and are mostly accepted by users, thereby help them to mitigate privacy disclosure in smartphones.

Index Terms—Mobile privacy, mobile applications, recommendation, crowdsourcing

1 Introduction

Mobile devices like smartphones or tablets are so popular today that billions of users all around the world are relying on them to handle personal and business affairs, like emails, calendar management, entertainment, etc. Unfortunately, the wide adoption of such devices are coming with some potential privacy threats, as they have gained access to lots of personal and sensitive data, such as user locations, contacts, and so on.

To mitigate such threats, system vendors have provided several mechanisms to confine the sensitive information accessible to mobile apps. For example, iOS from Apple has menu entries that enable users to control whether an app could access certain sensitive data sources. For Android, one of the most popular mobile platforms, its latest version (i.e., Android M released in May 2015) has similar fine-grained permission control mechanisms to replace its previous ineffective "all-or-none" scheme [1].

However, such a fine-grained control framework has its own drawbacks. For example, not all users have enough background knowledge to make the privacy configuration correctly. Also, there are so many apps and different permissions that it is really a tedious and challenging job for users to set all of them up. Finally, users hold different attitudes to the privacy, so there is no simple rule that can fit all demands. Some may be willing to provide some information for better services and experiences, while others may be reluctant to share sensitive data due to privacy concern. To achieve the best trade-off for each user, it is significant and beneficial to understand their expectations of privacy and help them to set the privacy permission accordingly.

In this paper, we propose a novel method that can help users establish their privacy settings properly and quickly. Our method is based on some key insights on how users decide whether to grant permission to an app or not. First, the decision depends on a user's specific privacy preference or concerns, for example, whether a user cares more about geographical location than contact lists. Second, the decision is also related to a user's expectations on certain apps, for example, a user would expect an alarm app to access calendar, but would not expect that app to access his/her current geographical location. Thus, the privacy permission should

- *R. Liu is with the School of Computer and Electronics Information, Guangxi University, Nanning 530004, China, and the CUHK Shenzhen Research Institute, Shenzhen 518057, China. E-mail: ruiliu@ie.cuhk.edu.hk.*
- *J. Liang is with the Guangxi Key Laboratory of Multimedia Communications and Network Technology, School of Computer and Electronics Information, Guangxi University, Nanning 530004, China. E-mail: liangjb@gxu.edu.cn.*
- *J. Cao and L. Yang are with the Department of Computing, Hong Kong Polytechnic University, Hong Kong. E-mail: {csjcao, csleiyang}@comp.polyu.edu.hk.*
- *K. Zhang is with the Department of Information Engineering, Chinese University of Hong Kong, Hong Kong. E-mail: khzhang@ie.cuhk.edu.hk.*
- *W. Gao is with the Department of Statistics, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061. E-mail: wenyu6@vt.edu.*
- *R. Yu is with the Software College, Northeastern University, Shenyang 110004, China. E-mail: yury@mail.neu.edu.cn.*

Manuscript received 17 Sept. 2015, revised 14 Sept. 2016, accepted 25 Nov. 2016, Date of publication 7 Dec. 2016; date of current version 5 Apr. 2019.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TSC.2016.2636285

be based on users' own preferences and expectations rather security experts' suggestions. More details and discussions will be provided in Section 2.

The method proposed in this paper is first to learn the similarities among users in terms of privacy preferences and privacy expectations on apps, and then to recommend appropriate permission settings to users based on such similarities. The rationale behind our method is that: users who share similar preferences on certain private data and/or privacy expectations on apps are more likely to make similar decisions in related privacy items.

To prove our proposed method, we have designed and implemented a system called *PriWe*, and evaluated it with lots of real world users (with 422 participants who finished our survey, and 78 recruited volunteers who provide feedback to our system). The results show that PriWe indeed is able to make proper recommendations that match users' privacy expectations and thus are mostly accepted by users.

Our Contributions. In this paper we make following contributions:

- We proposed PriWe to understand users' expectations of privacy on mobile apps using the crowdsourcing mechanism.
- We proposed a novel recommendation approach, combining the item-based and user-based collaborative filtering methods for generating the recommendations for users' privacy permission settings.
- We implemented and deployed PriWe in the real world for evaluation. We collected the feedback of 78 users who evaluated our system and 422 participants who completed our investigations. 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.

2 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 [2]. 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 [3]. These two acknowledged definitions both emphasized that privacy to people should include the ability to express themselves selectively. The expression is driven by the individual expectation of privacy.

Another conceptual framework for understanding privacy expectations, called contextual integrity, suggests that privacy comprises appropriateness and distribution [4]. More specifically, appropriateness focuses on whether the revelation of a particular piece of information is appropriate in a given context. For example, users' location data can be proffered in a map app rather than in a game app [5]. On the contrary, distribution defines the occurrence of an information transfer from one party to another. For instance, one person is willing to share his/her data with friends instead of strangers. Therefore, there is a trade-off between services and privacy. People's expectation of privacy is just a reflection of such trade-offs.

Using mobile apps is a typical scenario due to the discussion. Mobile devices, especially smartphones, have become an important platform which can provide multifarious services [6], [7], [8], [9]. There is almost no way to 100 percent protect users' information when they are using smartphones. More importantly, users also have ambivalent attitudes towards the data usage of mobile apps. They want to provide their data selectively based on their privacy expectations over mobile apps. On the one hand, we yearn for better services and performances so that we are willing to provide some information. On the other hand, in general we are reluctant to share information because we also hope that our sensitive data could be preserved. Thus, understanding the users' privacy expectation on mobile apps is a key point for addressing the privacy issues.

Furthermore, there is a significant difference between our work and the existing ones from the security perspective [10]. More specifically, security perspectives assume that there are correct options for the privacy permission settings or other security issues. They believe the opinions from security experts should be more useful and valuable as well. However, our pursuit is the balance between usability and privacy. There is no absolute right answer in our design and all the privacy permission settings are based on the users' expectations. Namely, we consider the people's own preference is appropriate even if it is against experts' suggestions. For example, some users do not understand security well and their settings are insecure according to experts' points of view. However, they may not care that much on security but want a more convenient usage of the apps instead. Then our recommendation will also sacrifice the security degree to add the usability accordingly. That is also the underpinning of our method and system.

3 System Design

Our previous paper proposed a prototype of this system [11]. In this section, we show the architecture of PriWe and elaborate on the mechanism we proposed to generate recommendations for privacy settings in smartphones.

3.1 Architecture

We have two intentions in our mind when designing PriWe. First, PriWe can help users to make better decisions on privacy settings in their own smartphones. Second, the processes of analyzing crowdsourced data and generating recommendations should be completed on a server due to the limited capability of smartphones. To achieve these intentions, we design the system, as illustrated in Fig. 1.

A mobile app is deployed to a smartphone to collect privacy settings from users. The mobile app of PriWe should consist of several features and provide various user-interfaces to interact with users. First, it can automatically scan the apps installed on the smartphone and identify them by names. The user can browse the privacy permission settings of each app accordingly. Second, the PriWe app can apply the recommendations generated by the server. For example, the app allows users to set/change the privacy permission of each mobile

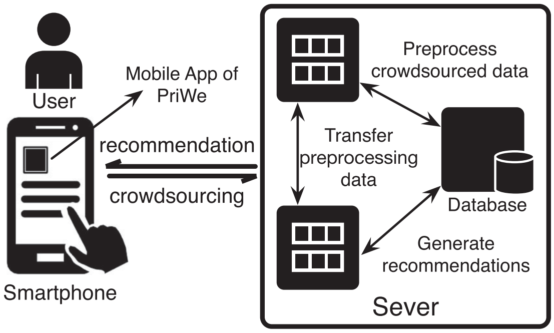


Fig. 1. The overview of PriWe, which insists of a mobile app and a server.

app installed on smartphones; it also can set the privacy permissions automatically when users attempt to apply the recommendations provided by the server. Finally, the PriWe app itself should hold the data access permissions as few as possible. Because the ultimate goal of our project is to help users make better decisions for privacy settings and mitigate potential privacy risk accordingly, our system should not be a privacy risk in any event.

The server side of PriWe has two key components, which are responsible for preprocessing the crowdsourced data and generating the recommendations, respectively. More specifically, the former components aim to preprocess the collected data, such as validation and classification; the latter one mainly focuses on generating recommendations for various mobile apps from different users. The proposed recommendation algorithm is deployed in this component. All the information, including the raw crowdsourced data and processed results, will be saved to an inbuilt database. The two components play a pivotal role on the server side, so we elaborate on them subsequently.

3.2 Recommendation Mechanism

In this section, we elaborate on the recommendation mechanism in PriWe. We present the reason we chose recommendation algorithm and the basic idea behind it in Section 3.2.1. Subsequently, we illustrate the item- and user-based collaborative filtering recommendation approaches in Section 3.2.2. Finally, we show how these two methods are fused to generate the recommendations of PriWe.

3.2.1 Basic Idea

Can you imagine what you will do, when you want to set the privacy permissions on your smartphone meanwhile you have no idea how to set them appropriately? One of the most immediate and intuitive thought is to ask someone or google it for others' suggestions. In this case, you actually want others to make some recommendations to you for setting the permissions. This is the reason why we adopt recommendation approaches to address the privacy issue. The basic idea for our work is that we want to, on behalf of you, collect the opinions from the people who have similar concerns with you.

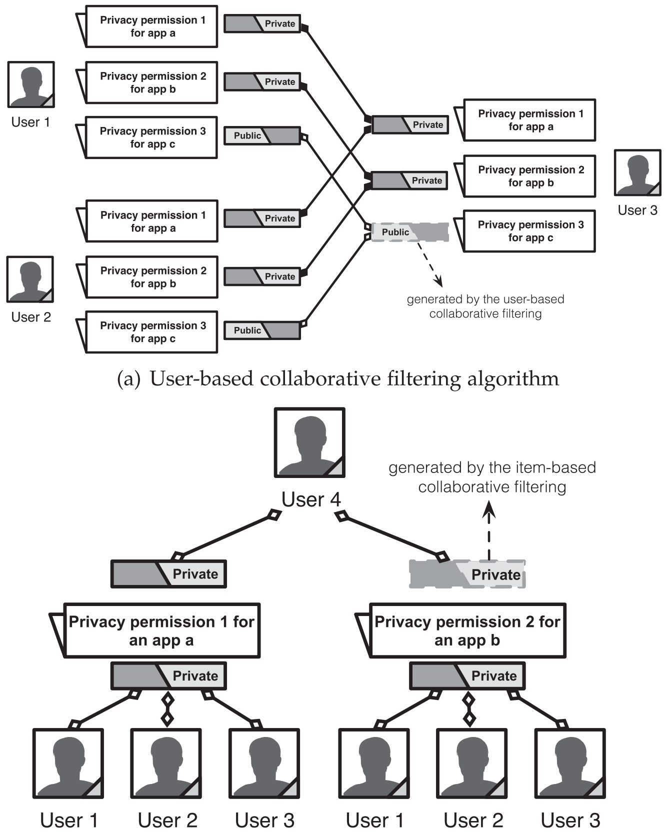
However, the traditional recommendation systems aimed to recommend attractive and interesting commodities to customers in some e-commerce markets, such as Amazon and Taobao. We do not have customers and commodities; rather we have smartphone users and privacy settings. We consider that the people with similar backgrounds, habits or ages may have similar privacy preferences. Thus, each user is mapped to a customer, and each privacy setting is mapped to a commodity. Therefore, the collaborative filtering algorithms can play an expected role in our work. Collaborative filtering technique has a long history and thrive recently. The two main categories are memory-based and model-based methods [12]. User- and item- based collaborative filtering are two key algorithms in memory-based methods. Model-based methods contain cluster-based CF, Bayesian classifiers, regression-based methods, and slope-one method, to name a few. There are also some state-of-the-art algorithms based on low-rank matrix factorization, such as regularized SVD method, Nonnegative Matrix Factorization, and Probabilistic Matrix Factorization, etc. Compared to the other two categories, memory-based algorithms have the following advantages : (1) They are nonparametric algorithms that depend less on the assumed model. Since we are studying from the data, we do not know the true model. Thus, any assumed model may restrict our usage of information. (2) They are easy to be popularized to higher dimensions. Although there are some techniques dealing with higher-order matrices, or tensors, matrix factorization can be very tedious and computationally heavy. On the contrary, user- and item-based algorithms are easy to compute and understand. (3) User-based algorithm is very robust on user count and item-based algorithm is robust on item count as well. Thus, our hybrid algorithm will be more robust on both user and item counts compared to other algorithms. Namely, our hybrid algorithm is more robust on sample size than other algorithms. (4) Despite of some parameters introduced to hybrid user- and item-based algorithms, our algorithm requires fewer parameters than the other algorithms. As a result, only a few parameters should be adjusted and the computation can be faster accordingly.

While all kinds of algorithms have their own pros and cons, in our situation, we think the above advantages of memory-based methods are more important to our interest. Thus, we only consider an improved algorithm regarding to user- and item-based algorithms, which are the two main algorithms in memory-based category. Therefore, to achieve better performances and overcome their intrinsic drawbacks, we combine the two collaborative filtering algorithms based on the conditional probability. The method can generate the recommendations for different persons according to the crowdsourced privacy permission settings.

According to the basic idea, our recommendation algorithm is initialized by the crowdsourced users' privacy permission settings rather than some experts' opinions. That is because we believe users' expectation should be the key to set the privacy permissions of their mobile apps.

3.2.2 Item- and User-Based Collaborative Filtering

We assume that there are K users, each user has M apps. Each app holds N data access permissions. We define $r_{i,a,g}$ as the setting of data permission g of the app a set by the user i. More specifically, the users set the privacy setting by the dichotomous variable {0, 1}, where $r_{i,a,g} = 0$ denotes that the users are averse to share the data with anyone, whereas $r_{i,a,g} = 1$ means the participant allows the disclosure of that information. However, the users may not have a clear understanding to various privacy permissions, and it is arduous for them to establish all of the privacy settings. The


Privacy permission 1 for app a
Privacy permission 2 for app b
Privacy permission 3 for app c
User 1
Private
Private
Public
Privacy permission 1 for app a
Privacy permission 2 for app b
Privacy permission 3 for app c
User 2
Private
Private
Public
Private
Private
Public
Privacy permission 1 for app a
Privacy permission 2 for app b
Privacy permission 3 for app c
User 3
generated by the user-based collaborative filtering
User 4
generated by the item-based collaborative filtering
Private
Private
Privacy permission 1 for an app a
Privacy permission 2 for an app b
Private
Private
User 1
User 2
User 3
User 1
User 2
User 3

(a) User-based collaborative filtering algorithm

(b) Item-based collaborative filtering algorithm

Fig. 2. Generating recommendation of data access permissions for Android apps is based on the user- and item-based collaborative filtering algorithm.

recommendations made by PriWe can help the users to set the data access permission if they have ambiguous understanding of the associated data. We propose a recommendation mechanism by drawing inspiration from the user-based and item-based collaborative filtering approaches, as portrayed in Fig. 2. The following two examples further illustrate these two approaches.

Example 1. Two users, i and j, both installed two apps a, b in the smartphone, and each app holds two permissions g, h. The user i and j both allow the app a to get the corresponding data permissions, by setting $r_{i,a,g} = 1$ & $r_{i,a,h} = 1$ and $r_{j,a,g} = 1$ & $r_{j,a,h} = 1$. In this situation, we consider that they may have similar privacy preferences. If the user i set $r_{i,b,g} = 0$ to prohibit the access permission g of the app b, then user j is likely to have the same choice on this setting.

Example 2. Two apps, a' and b', both are installed in the smartphone carried by the user i' and user j'. The apps a' and b' hold the permissions g' and h', respectively. If the users i' and j' both reject the data access, namely setting the $r_{i',a',g'} = 0$ & $r_{i',b',h'} = 0$ and $r_{j',a',g'} = 0$ & $r_{j',b',h'} = 0$. In this case, the permission g' of app a' and permission h' of app b' can be considered as two similar ones because they are both rejected by the users i' and j'. The more users, the higher similarity. Thus, when newcomers have the negative opinion to the privacy permission g' of app a', we also recommend them to reject the data access of permission h' of app b'.

The examples illustrate the user-based and item-based collaborative filtering approaches. Based on the examples, the key of applying the collaborative filtering approaches is to find similar apps and the people who have the similar privacy preferences. Thus, we calculate the similarity of users and apps, respectively.

We define $s_u(i,j)$ as the similarity between the user i and j. The similarity reflects how similar the users i and j are, particularly, how many of the same privacy settings the two users have. 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 1 indicate a strong similarity. 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 who have similar basic information may have similar privacy preferences. PriWe will first categorize the users into different groups due to their backgrounds, habits and ages. Then, PriWe calculates the similarity between the users in the same group and those in the different group afterwards

$$s_u(i,j) = \frac{\sum_{a\in M}\sum_{g\in N}(r_{i,a,g} - \overline{r}_i)(r_{j,a,g} - \overline{r}_j)}{\sqrt{\sum_{a\in M}\sum_{g\in N}(r_{i,a,g} - \bar{r_i})^2}\sqrt{\sum_{a\in M}\sum_{g\in N}(r_{j,a,g} - \bar{r_j})^2}}. \tag{1}$$

We obtain the set of similar users by applying a threshold or $top - N$ strategy. The $top - N$ set of similar users for user i, $S_u(i)$ can be generated according as

$$S_u(i) = \{j|rank\ s_u(i,j) \leq N\}. \tag{2}$$

Likewise, we define $s_i(g,h)$ as the similarity between the privacy permission g and h. The similarity is based on the existing users' settings as illustrated in the Example 2. To calculate the similarity, we adopt the adjusted cosine similarity to take the differences of the average setting behaviors of the users into account, as shown in Eq. (3). We also select $top - N$ similar items according to Eq. (4)

$$s_i(g,h) = \frac{\sum_{i\in K}\sum_{a\in M}(r_{i,a,g} - \overline{r}_i)(r_{i,a,h} - \overline{r}_i)}{\sqrt{\sum_{i\in K}\sum_{a\in M}(r_{i,a,g} - \overline{r}_i)^2}\sqrt{\sum_{i\in K}\sum_{a\in M}(r_{i,a,h} - \overline{r}_i)^2}} \tag{3}$$

$$S_i(g) = \{h|rank\ s_i(g,h) \leq N\}. \tag{4}$$

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 [13]. However, it has been presented that the adjusted cosine similarity consistently outperforms the Pearson correlation metric in the item-based recommendation situations.

3.2.3 Putting It All Together

After calculating the similarities $s_u(i,j)$ and $s_i(g,h)$, we propose a probabilistic-based similarity fusion framework and adopt it to generate a more robust similarity and overcome the data sparsity problem, which is an obstacle to our work in real-world deployment. The basic idea is that we provide different weights to the two similarities $s_u(i,j)$ and $s_i(g,h)$ based on the probability, and combine them accordingly. Thus, the user-based and item-based collaborative filtering approaches are only two special cases in the unified algorithm. We elaborate on the algorithm in the remainder of the section.

Assume we want to make a recommendation for user x about the privacy setting of permission z of app y in his smartphone, namely obtaining $r_{x,y,z}$. Due to the aforementioned discussion, user-based recommendation algorithm only considers privacy settings provided by similar users. We define the existing privacy settings for calculating $r_{x,y,z}$ as a set, US, where $US_{x,y,z} = \{r_{i,y,z} | i \in S_u(x)\}$. Likewise, item-based recommendation considers privacy settings from similar items. We also define a set, IS, where $IS_{x,y,z} = \{r_{x,a,g} | g \in S_i(z)\}$. The similarity fusion algorithm considers these two sets jointly, i.e., UIS, where $UIS_{x,y,z} = \{r_{i,a,g} | i \in S_u(x), g \in S_i(z), x \neq i, z \neq g\}$.

When we scrutinize the crowdsourced privacy settings from different users, we find people have really different preferences to the same permission due to their intrinsic traits. To decrease the effect, we first normalize the collected privacy settings by removing the average values, as shown as

$$p_{x,y,z}(r_{i,a,g}) = r_{i,a,g} - (\overline{r}_i - \overline{r}_x) - (\overline{r}_{a,g} - \overline{r}_{y,z}). \tag{5}$$

The $p_{x,y,z}(r_{i,a,g})$ serves as a normalized function of the privacy setting of the permission z of the app y set by the user x, based on the existing crowdsourced privacy setting $r_{i,a,g}$. $\overline{r}_{a,g}$ and $\overline{r}_{y,z}$ are the mean of the privacy setting of the permission g of the app a and the privacy setting of the permission z of the app y, respectively.

We define a sample space of the privacy settings as $\Phi_r = \{\emptyset, 0, 1, 2, \ldots, r\}$ without loss of generality. In our case, there are actually three options, $\{\emptyset, 0, 1\}$. The $\emptyset$ denotes the unknown privacy settings, 0 means users regard the information as private, and 1 presents that users allow the disclosure of that information. Let $r_{i,a,g}$ denote a crowdsourced privacy setting of the permission g of the app a, which is provided by the user i, over the sample space Φ_r. Then we define $\mathbb{P}(r_{x,y,z}|\Omega_{x,y,z})$ as a conditional probability of the decision made by user x on permission z of app y, given a set of normalized settings, $\Omega_{x,y,z}$, where

$$\Omega_{x,y,z} = \{p_{x,y,z}(r_{i,a,g}) | r_{i,a,g} \neq \emptyset\}. \tag{6}$$

Thus, considering the set of normalized settings based on user-based similarity and item-based similarity (i.e., $r_{i,a,g} \in US_{x,y,z} \cap IS_{x,y,z}$), we can get the conditional probability in the light of the normalized privacy settings from the set $US_{x,y,z}$ and $IS_{x,y,z}$ as presented

$$\begin{aligned}&\mathbb{P}(r_{x,y,z}|\Omega_{x,y,z})\\ &= \mathbb{P}(r_{x,y,z}|US, IS)\\ &= \mathbb{P}(r_{x,y,z}|\{p_{x,y,z}(r_{i,a,g}) | r_{i,a,g} \in US \cup IS\}).\end{aligned} \tag{7}$$

We introduce two independent binary indicators I_1 and I_2 to present the dependency on set US and IS. That is, $I_1 = 1$ corresponds to dependency on the set US while $I_1 = 0$ indicates independency. Likewise, $I_2 = 1$ states $r_{x,y,z}$ depends on the set IS while $I_2 = 0$ indicates $r_{x,y,z}$ is independent of IS. Therefore, given the two sets US and IS, we can derive Eq. (8) based on the indicators I_1 and I_2

$$\begin{aligned}&\mathbb{P}(r_{x,y,z}|US, IS)\\ &= \sum_{I_1}\sum_{I_2} \mathbb{P}(r_{x,y,z}|I_1, I_2, US, IS)\mathbb{P}(I_1, I_2|US, IS)\\ &= \mathbb{P}(r_{x,y,z}|I_1 = 0, I_2 = 0, US, IS)\mathbb{P}(I_1 = 0, I_2 = 0|US, IS)\\ &+ \mathbb{P}(r_{x,y,z}|I_1 = 1, I_2 = 0, US, IS)\mathbb{P}(I_1 = 1, I_2 = 0|US, IS)\\ &+ \mathbb{P}(r_{x,y,z}|I_1 = 0, I_2 = 1, US, IS)\mathbb{P}(I_1 = 0, I_2 = 1|US, IS)\\ &+ \mathbb{P}(r_{x,y,z}|I_1 = 1, I_2 = 1, US, IS)\mathbb{P}(I_1 = 1, I_2 = 1|US, IS).\end{aligned} \tag{8}$$

According to the definition of indicators I_1, I_2, $r_{x,y,z}$ is independent from US when $I_1 = 0$ and is irrelevant to IS if $I_2 = 0$. Thus, $\mathbb{P}(r_{x,y,z}|I_1 = 1, I_2 = 0, US, IS) = \mathbb{P}(r_{x,y,z}|US)$, $\mathbb{P}(r_{x,y,z}|I_1 = 0, I_2 = 1, US, IS) = \mathbb{P}(r_{x,y,z}|IS)$. Moreover, we cannot generate any recommendation without the sets US and IS, which means $\mathbb{P}(r_{x,y,z}|I_1 = 0, I_2 = 0, US, IS) = 0$. When we consider the sets US and IS jointly, these two sets can be regarded as the set UIS. Namely, $\mathbb{P}(r_{x,y,z}|I_1 = 1, I_2 = 1, US, IS) = \mathbb{P}(r_{x,y,z}|UIS)$. Therefore, we can obtain

$$\begin{aligned}&\mathbb{P}(r_{x,y,z}|US, IS)\\ &= \mathbb{P}(r_{x,y,z}|US)\mathbb{P}(I_1 = 1, I_2 = 0|US, IS)\\ &+ \mathbb{P}(r_{x,y,z}|IS)\mathbb{P}(I_1 = 0, I_2 = 1|US, IS)\\ &+ \mathbb{P}(r_{x,y,z}|UIS)\mathbb{P}(I_1 = 1, I_2 = 1|US, IS).\end{aligned} \tag{9}$$

For easy computation, we introduce two parameters λ and δ in Eq. (10), assuming $\mathbb{P}(I_1 = 1|US, IS) = \lambda$ and $\mathbb{P}(I_2 = 1|US, IS) = \delta$. According to the Eq. (10), the $r_{x,y,z}$ depends on both set US and IS, namely UIS, when $\lambda = 1$ and $\delta = 1$. Likewise, the $r_{x,y,z}$ has 0.5 probability dependent on US, if $\lambda = 0.5$; the set IS also can play a half role when δ is set to 0.5

$$\begin{aligned}&\mathbb{P}(r_{x,y,z}|US, IS)\\ &= \mathbb{P}(r_{x,y,z}|US)\lambda(1-\delta)\\ &+ \mathbb{P}(r_{x,y,z}|IS)(1-\lambda)\delta + \mathbb{P}(r_{x,y,z}|UIS)\lambda\delta.\end{aligned} \tag{10}$$

Afterwards, we can get the estimated privacy settings $r_{x,y,z}$, as presented in Eq. (11). We can determine the parameter λ and δ through iteration in the experiments

$$\begin{aligned}\widehat{r}_{x,y,z} &= \sum_{t=1}^{\Phi_r} t\mathbb{P}(r_{x,y,z} = t|US, IS)\\ &= \left(\sum_{t=1}^{\Phi_r} t\mathbb{P}(r_{x,y,z} = t|UIS)\lambda\delta\right) +\\ &\left(\sum_{t=1}^{\Phi_r} t\mathbb{P}(r_{x,y,z} = t|US)\lambda(1-\delta)\right) +\\ &\left(\sum_{t=1}^{\Phi_r} t\mathbb{P}(r_{x,y,z} = t|IS)(1-\lambda)\delta\right).\end{aligned} \tag{11}$$

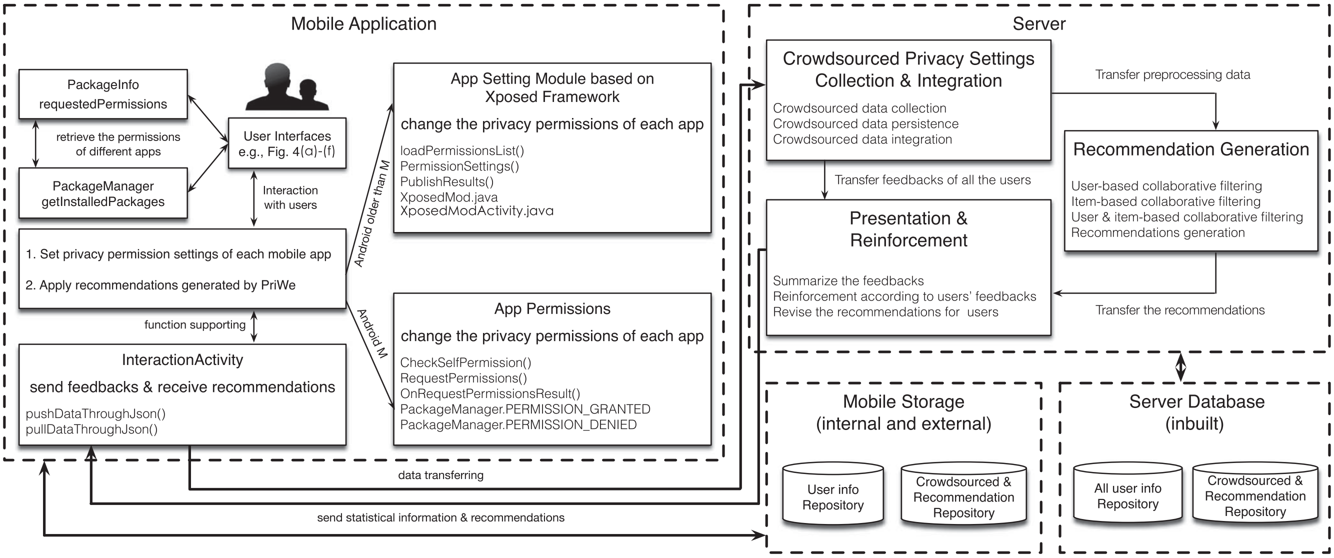

Mobile Application
PackageInfo
requestedPermissions
retrieve the permissions of different apps
PackageManager
getInstalledPackages
User Interfaces
e.g., Fig. 4(a)-(f)
Interaction with users
1. Set privacy permission settings of each mobile app
2. Apply recommendations generated by PriWe
function supporting
InteractionActivity
send feedbacks & receive recommendations
pushDataThroughJson()
pullDataThroughJson()
Android older than M
Android M
App Setting Module based on Xposed Framework
change the privacy permissions of each app
loadPermissionsList()
PermissionSettings()
PublishResults()
XposedMod.java
XposedModActivity.java
App Permissions
change the privacy permissions of each app
CheckSelfPermission()
RequestPermissions()
OnRequestPermissionsResult()
PackageManager.PERMISSION_GRANTED
PackageManager.PERMISSION_DENIED
Server
Crowdsourced Privacy Settings Collection & Integration
Crowdsourced data collection
Crowdsourced data persistence
Crowdsourced data integration
Transfer preprocessing data
Recommendation Generation
User-based collaborative filtering
Item-based collaborative filtering
User & item-based collaborative filtering
Recommendations generation
Transfer feedbacks of all the users
Presentation & Reinforcement
Summarize the feedbacks
Reinforcement according to users' feedbacks
Revise the recommendations for users
Transfer the recommendations
Mobile Storage (internal and external)
User info Repository
Crowdsourced & Recommendation Repository
Server Database (inbuilt)
All user info Repository
Crowdsourced & Recommendation Repository
data transferring
send statistical information & recommendations

Fig. 3. The implementation architecture of PriWe.

Now we need to estimate the conditional probability in Eq. (11), namely, $\mathbb{P}(r_{x,y,z}=t|UIS)$, $\mathbb{P}(r_{x,y,z}=t|US)$, and $\mathbb{P}(r_{x,y,z}=t|IS)$. The basic idea of the estimation is to calculate the likelihood of $r_{x,y,z}$ to be similar to $r_{i,a,g}$ based on the sets US, IS, and UIS. Hence, we make use of the similarity between users to calculate the likelihood based on US, as shown in Eq. (12). Likewise, the similarity function $s_i(.)$ is used to compute the likelihood based on the set IS, as presented in Eq. (13)

$$\mathbb{P}(r_{x,y,z}=t|US) = \frac{\sum_{\forall r_{i,a,g}:(r_{i,a,g}\in US_{x,y,z})\wedge(p_{x,y,z}(r_{i,a,g})=t)} s_u(i,x)}{\sum_{\forall r_{i,a,g}:r_{i,a,g}\in US_{x,y,z}} s_u(i,x)} \tag{12}$$

$$\mathbb{P}(r_{x,y,z}=t|IS) = \frac{\sum_{\forall r_{i,u,a}:(r_{i,a,g}\in IS_{x,y,z})\wedge(p_{x,y,z}(r_{i,a,g})=t)} s_i(g,z)}{\sum_{\forall r_{i,a,g}:r_{i,a,g}\in IS_{x,y,z}} s_i(g,z)}. \tag{13}$$

Calculating the likelihood based on UIS is a little tricky. We consider the probability estimation as the combination of the similarity function $s_u(.)$ and $s_i(.)$. More specifically, we use euclidean distance to produce the similarity function, as illustrated in Eq. (15)

$$\mathbb{P}(r_{x,y,z}=t|UIS) = \frac{\sum_{\forall r_{i,a,g}:(r_{i,a,g}\in UIS_{x,y,z})\wedge(p_{x,y,z}(r_{i,a,g})=t)} s_{ui}(r_{i,a,g},r_{x,y,z})}{\sum_{\forall r_{i,a,g}:r_{i,a,g}\in UIS_{x,y,z}} s_{ui}(r_{i,a,g},r_{x,y,z})} \tag{14}$$

$$s_{ui}(r_{i,a,g},r_{x,y,z}) = \frac{1}{\sqrt{\left(\frac{1}{s_u(i,x)}\right)^2+\left(\frac{1}{s_i(g,z)}\right)^2}}. \tag{15}$$

Now, we can get the results in Eq. (16), combining the above estimations of conditional probability

$$\widehat{r}_{x,y,z} = \sum_{r_{i,a,g}} p_{x,y,z}(r_{i,a,g}) W^{i,a,g}_{x,y,z}, \tag{16}$$

where

$$W^{i,a,g}_{x,y,z} = \begin{cases} \frac{s_u(i,x)}{\sum_{r_{i,a,g}\in US} s_u(i,x)}\lambda(1-\delta) & r_{i,a,g}\in US \\ \frac{s_i(g,z)}{\sum_{r_{i,a,g}\in IS} s_i(g,z)}(1-\lambda) & \delta r_{i,a,g}\in IS \\ \frac{s_{ui}(r_{i,a,g},r_{x,y,z})}{\sum_{r_{i,a,g}\in UIS} s_{ui}(r_{i,a,g},r_{x,y,z})}\lambda\delta & r_{i,a,g}\in UIS. \end{cases} \tag{17}$$

So far, we have elaborated on the process of recommendation based on the crowdsourced privacy settings. The only thing is to determine the parameters λ and δ. When we deploy the system in the real world, we find these two parameters, λ and δ, reach their optimal at 0.7 and 0.5, respectively. According to the illustration of algorithm, the parameters are determined by the dataset, which means they are adaptive. More details are presented in Section 5.3.

4 Implementation

In this section, we describe the implementation of PriWe, including both the app side and server side. The implementation architecture of PriWe is illustrated in Fig. 3.

4.1 PriWe App

Our prototype of PriWe app is implemented on Google/LG Nexus 4 handset running with Android version 4.4. It has been tested on other real world Android devices with Android versions ranging from 4.0.3 to 5.1.1.

There are two major objectives when designing the PriWe mobile app. First, it should provide some user input elements that enable users to set or change permission settings related to privacy. Such settings will then be sent back to server for generating recommendations. Second, the PriWe app should be able to apply the recommendation results received from server. It can set the privacy permissions automatically when a user has confirmed to take the recommended settings. It also provides some extra features to improve user experiences. As shown in Fig. 3, users can browse what apps have been installed on their smartphones, and what data access permission the apps have

TABLE 1
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 subscribing 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.)

been granted. All of these are done with help from standard APIs in official Android SDK. For example, function *PackageManager.getInstalledPackages(0)* can retrieve installed apps in the smartphone. Function *PackageInfo.requestedPermissions* can scrutinize the privacy permissions of each app. Such functions can return all the granted permissions of each app for users to review. However, it is arduous for users to read all the permissions with such a small screen of smartphone, so we summarize eleven types of abused data and permissions of Android apps and discuss their potential risks, as shown in Table 1. The summary is based on some freeform comments on the Internet [14], [15], research papers on the Android system and analysis of smartphone apps [16], [17], [18], [19], [20], security and privacy tips from official guidelines [21], and a survey on information security and privacy of Android apps [22].

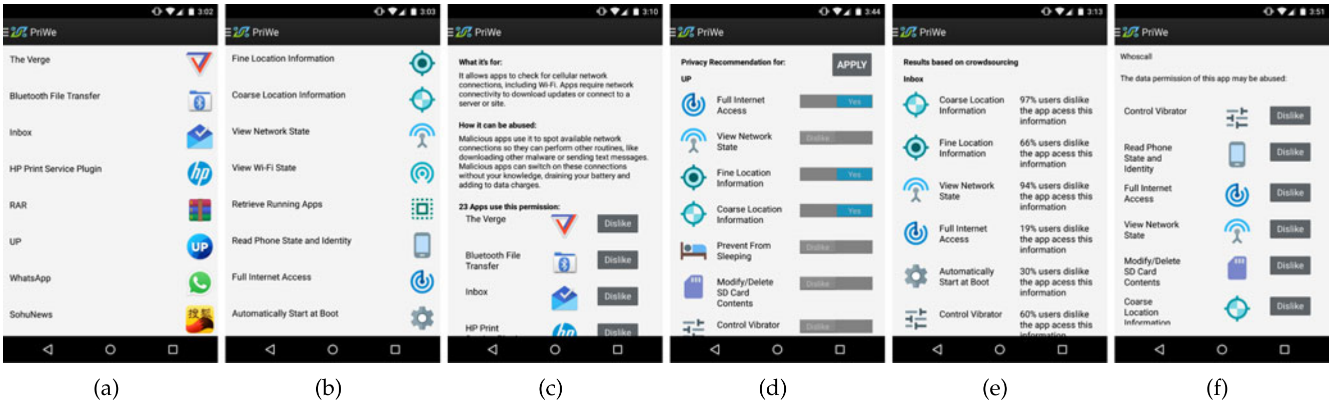
In order to change the permission settings, the PriWe app needs root privilege or to run as a system level process. However, it should not be a problem, as the methods proposed in this paper is expected to be adopted by Android system vendors and device manufacturers like Google and Samsung, and any built-in app from such companies can easily gain root privileges. For security reasons, we do not advocate users to root their smartphones which could open more attacking surfaces.

There are two different approaches to apply the recommended permission settings, depending on Android OS versions. For devices with OS versions before 5.0, PriWe app can use the App Setting Module, which is based on Xposed Framework [23] since such Android systems did not provide any mechanism for normal users to modify the privacy settings. For devices with Android 5.0 (i.e., Android M) or later, PriWe app will invoke the functions provided by the Android OS to change the permissions of each app.

The key functions of PriWe mobile 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, which are also provided by Android SDK.

4.2 PriWe Server

The server is designed to analyze the collected data and generate recommendations accordingly. As shown in Fig. 3, there are three key components in the server, which are responsible for data preprocessing, recommendation generation, and presentation and reinforcement. In the data preprocessing part, the server mainly focuses on cleaning and structuring the collected data, which will be the input of the



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)

Fig. 4. PriWe provides an Android app for participants. (a) PriWe can scan various apps 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.

next step, i.e., the recommendation generation. To generate recommendations, the server applies the proposed method depicted in Section 3.2. The output of this step is a list of privacy permission settings. In the presentation and reinforcement part, the server can summarize the collected data and provide users some statistics information. The reinforcement part will revise the recommendation list if the user changed the permission settings even after accepting the recommendations.

The server system is deployed on an IBM server and has a typical three-tier architecture, i.e., application tier, a domain logic tier, and a data persistence tier. More specifically, the application tier is the Web front-end implemented with HTML, JavaScript and third-party libraries, providing a user friendly interface. The domain logic tier is implemented with 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 on frameworks including Spring, Struts, and Hibernate. The recommendation algorithm is put to this tier. In the data persistence tier, all data are stored in a MySql database.

5 Evaluation

In this section, we will present evaluation results of PriWe, including experiments setup, data collection methods, evaluation outputs and our findings. We have conducted two experiments: one is based on users' survey, and the other one is based on users' experience of real world deployment. During the evaluation, we will use a metric defined in Eq. (18) to measure the accuracy (or effectiveness) of proposed recommendation algorithm, where R_p denotes all the privacy permission settings the participants have chosen in the experiment, and R_i represents the recommendations of the corresponding privacy permission settings provided by PriWe

$$Accuracy(i) = \frac{R_p \cap R_i}{R_i}. \tag{18}$$

5.1 Evaluation Based on Survey

We published a task on the Amazon Mechanical Turk[1] for three weeks, and 382 participants completed our task. In the task, we asked the participants to answer a questionnaire to indicate their privacy preferences about various types of mobile apps. In order to get a better understanding, we prepared two questionnaires, survey A and survey B. Survey A is used to get the privacy preferences of participants towards various apps widely, while survey B is used to collect fine-grained participants' preferences on certain privacy permission requested by some particular mobile apps. 200 participants completed the survey A and 182 participants finished the survey B.

We have performed some statistical analysis on the background of all participants, and found that they are distributed in terms of age, gender, work/professional background, and use habit. 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, finance, information technology, etc. To further avoid bias of the population in Amazon Mechanical Turk, we intentionally recruited 40 female participants from non-IT areas in the real world to complete the survey A and B, the same as the participants did in Amazon Mechanical Turk. They also vary on nationality, age and occupation. Thus, we believe that the population of the participants is more or less evenly distributed and our data are unbiased and the analysis results should be convincing accordingly. All the information about the distribution of the participants in survey A and survey B can be shown in the Table 2.

To evaluate the accuracy of recommendations produced by PriWe, we followed the standard practice by splitting the data into two sets: one for training and the other for testing. This is done at multiple granularity levels in order to get more complete and cross-verified results. For example, we split data from survey A into two sets and perform the tests, and the same is done on survey B. We also used the whole survey A data as training set and survey B data as test set, and vice versa.

All the results are demonstrated in Fig. 5. The overall accuracy of the recommendations made by PriWe is about 79 percent. It means that most recommendations are accurate and appropriate and thus have been accepted by users. According to the results, the results based on survey A and survey B jointly are better than those based on either survey A or survey B. It indicates the recommendations can achieve higher accuracy when the data set consists of more crowdsourced permission settings. The combination of two surveys can also overcome the data sparsity issues in some degree.

We presented the results according to participants' gender, age, background, time spent on smartphones, favourite activity on smartphones and attitude to the survey, as shown in Figs. 5a, 5b, 5c, 5d, 5e, and 5f. Fig. 5a demonstrates the recommendations provided by PriWe for male participants can achieve slightly higher accuracy than those for females. There is no obvious evidence to support that males have better understanding to the privacy permission of mobile apps. However, what we find is that female participants spend most of the time on the smartphone shopping and socializing. It may suggest that female users do not have enough attentions on the personal information on the smartphone. Another finding is that accuracy becomes higher gradually with the increase of participants' ages. One potential explanation is that some young people have no unambiguous perceptions about their privacy permission of their mobile apps. For the participants who have information technology background, the accuracy of recommendation for the participants with a focus on privacy and security (around 90 percent) is higher than the remainder of all the selected participants. This may be because their experience about information privacy and security improve their awareness and help them to make an appropriate choice about the privacy permission settings in smartphones. Due to the same reason, the users who came from other areas have relatively lower accuracy of recommendations. Fig. 5d indicates PriWe did not provide so proper advices to the

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

TABLE 2
Statistics of Participants in Survey

Participants	Numbers in Survey A	Percentage in Survey A	Numbers in Survey B	Percentage in Survey B
Male	133	55.4%	110	49.5%
Female	107	44.6%	112	50.5%
10-19	9	3.8%	11	5%
20-24	45	18.8%	43	19.4%
25-29	69	28.8%	70	31.5%
30-40	84	35%	71	32%
40+	33	13.8%	27	12.2%
Energy	9	3.8%	6	2.7%
Materials	14	5.8%	16	7.2%
Industrials	19	7.9%	22	9.9%
Consumer Discretionary	23	9.6%	17	7.7%
Consumer Staples	22	9.2%	27	12.2%
Health Care	24	10%	17	7.7%
Finance	28	11.7%	21	9.5%
IT in Security & Privacy	27	11.3%	25	11.3%
IT in non Security & Privacy	40	16.7%	39	17.6%
Tele Services	15	6.3%	19	8.6%
Utilities	19	7.9%	13	5.9%
Rarely (0 1hr)	11	4.6%	13	5.9%
Sometimes (1 2hr)	56	23.3%	56	25.2%
Frequently (2 4 hr)	98	40.8%	75	33.8%
Very often (4+ hr)	75	31.3%	78	35.1%
Socializing	86	35.8%	67	30.2%
Shopping	33	13.8%	26	11.7%
Accomplishing	12	5%	16	7.2%
Arrangement	15	6.3%	17	7.7%
Discovery	30	12.5%	27	12.2%
Me Time	50	20.8%	44	19.8%
Self-expression	14	5.8%	25	11.3%
Deliberately completed	149	62.1%	155	69.8%
Normally completed	84	35%	65	29.3%
Hastily completed	7	2.9%	2	0.9%

users who spent less time on the smartphone. Such users may have inadequate knowledge to the devices which they did not take much time. As shown in Fig. 5e, people who like to use some accomplishing (e.g., managing finances, health and productivity) or arrangement (planning for upcoming events) apps will get more accurate recommendations from PriWe due to their existing and crowdsourced permission settings. In the last subfigure Fig. 5f, we can see the people who completed our task in a rush cannot get proper recommendations for their privacy permission settings since they just finish the task without any attention.

To articulate the performance of our method, we implemented a naive heuristic method as baseline. We also compared our work with item-based collaborative filtering and user-based collaborative filtering, respectively, to highlight the advantages of the proposed method. The basic idea of the naive method is: 1) classify the apps according to the taxonomy provided by Google Play Store, 2) predefine some rational granted permissions based on common sense, 3) accept all the permission requests if they belong to the predefined permissions group, otherwise reject them. This naive method can reflect the basic common idea about permission settings in a sense, which is regarded as the baseline in our evaluation. It is very arduous to go over all the permissions of all the apps to generate a completed rational permissions group, nevertheless we try our best to check all the popular apps [15] and most common permission requests [14], [24]. For user- and item-based collaborative filtering, we implemented these two methods and applied them on the collected data. Like the aforementioned evaluation, we separated survey A dataset into two parts (one is regarded as training data and the other one is testing data), and the same was done on survey B dataset. We jointly considered survey A and survey B, training the algorithm by survey A dataset and testing it via survey B dataset and vice versa. Moreover, we also tested these three algorithms on the dataset of participants whose background involved IT security and privacy to figure out whether users will follow the experts' opinions. Note that the naive method did not require training process and it is deployed to generate the permission settings based on the aforementioned strategy accordingly. The results are presented in Fig. 6. According to the evaluation, our proposed method always has the overwhelming superiority in comparison with baseline and the performance of item-based collaborative filtering is better than that of user-based collaborative filtering. Compared with these two methods, our method can generate a more robust similarity and overcome the data sparsity problem. The results indeed corroborate these advantages. Moreover, the recommendation accuracy based on the experts'

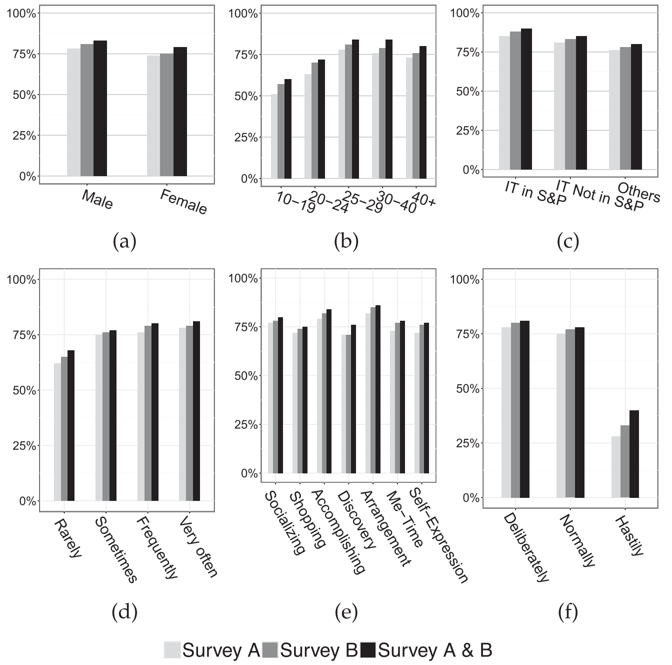


Fig. 5. The accuracy of recommendation generated by PriWe based on the participants' feedbacks in Amazon Mechanical Turk. The results are presented according to (a) the participants' genders (b) the participants' ages (c) the participants' backgrounds (d) the time participants spent on the smartphone (e) the most frequent activities of participants and (f) the attitudes of participants.

suggestions is much lower than the three previous ones. Should be noted that our algorithm is independent of initial dataset, i.e., our algorithm still can obtain acceptable accuracy eventually (after collecting more normal users's data), even the algorithm is initialized by experts' opinion. We just test the influence of experts' opinions so we did not launch the evaluation for a long time. The result shows that experts always take the security and privacy concern as the first priority rather than usability. However, normal users' hope to find a balance between privacy and usability instead of sacrificing any of them. Even the security experts' suggestion can protect users' privacy in a sense, the users still did not accept them due to usability of apps. Thus, the acceptance rate of our recommendation cannot be high when we only take experts' opinion as the root of method.

We also further conducted *K-fold cross validation* to present generalization of our performance, by comparison with user-based and item-based collaborative filtering. Although in general K remains an unfixed parameter, 10-fold cross validation is commonly used [25]. Following the convention, we set $K = 10$, which means we separated the dataset into 10 parts, tested each of them using the rest. According to Fig. 7, PriWe on each testing dataset can achieve stable and high accuracy. Compared with other two collaborative filtering methods, our method always can obtain better results.

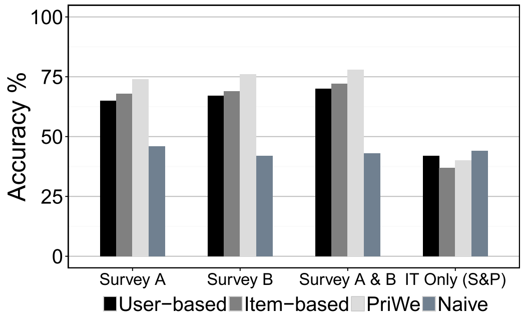


Fig. 6. Comparison between user-based collaborative filtering, item-based collaborative filtering, our proposed method PriWe, and naive method (baseline), based on data from survey A, survey B, survey A & B, and people whose background involved security and privacy.

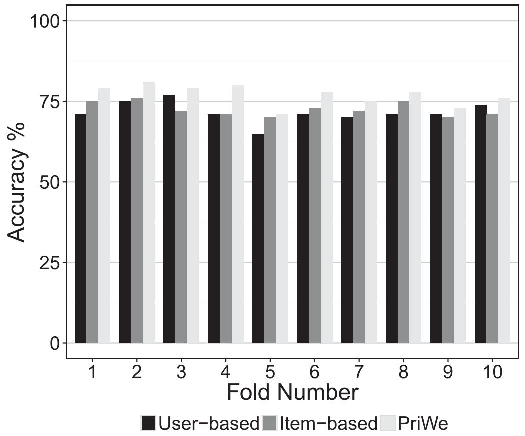


Fig. 7. K-fold cross validation on collected dataset (K=10).

5.2 Evaluation Based on Real-World Deployment

PriWe app is released to 78 users, who are from Hong Kong, Singapore, Austria, England, America and China, and are volunteered to evaluate PriWe's performance in the real world. The server collected users' feedback, including their permission settings of installed apps, as well as some basic information like gender, age, unique user ID, etc. Since users have multifarious apps, the summary of number apps of each user is shown in Table 3. The data showed that most users have less than 40 apps installed on their smartphones, which roughly matches the data from Statistics Portal [26].

To corroborate the proposed abused data and permissions list, we calculated the average number of Android apps that participants installed that access these data and permissions. According to the results as presented in Table 4, we found that all the potential abused privacy permissions have really been requested by many apps. These apps account for a large proportion of all the apps in light of Table 3.

Since there is no open dataset or existing metric to evaluate our work, and the above survey dataset is coming from real world users, we treat it as the ground truth to evaluate PriWe.

We illustrated the evaluation results in Figs. 8 and 9 respectively. Fig. 8 shows the percentage of PriWe recommendations taken by participants. The result indicates that most recommendations are taken by the users, except two

TABLE 3
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%

TABLE 4
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

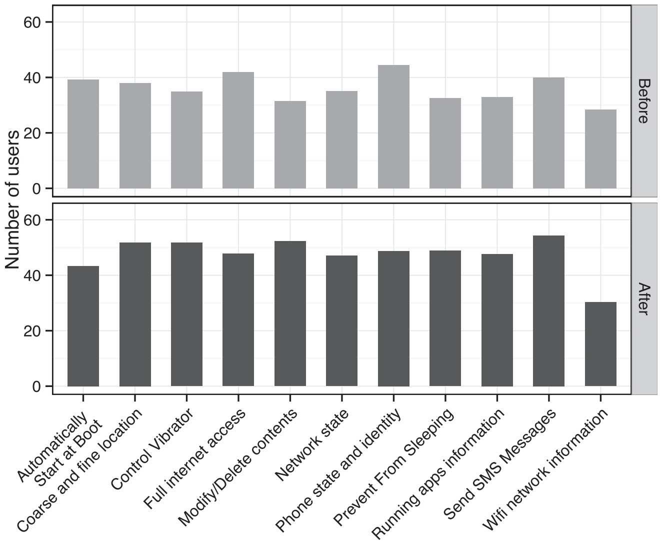


Fig. 9. The number of users have a better understanding of each data access permission after using PriWe.

permissions of *Prevent From Sleeping* and *Control Vibrator*. Based on user feedbacks, one possible reason might be that experiment participants do not fully understand the meaning of these two permissions, especially for the potential security risks, thus had ignored PriWe recommendations. In contrast, other highly risky permissions, like *Coarse and fine location* and *Automatically Start at Boot*, have more obvious security implications that can be easily captured by participants, thus the recommendations from PriWe were highly accepted. For permissions that could be risky but at the same time may be critical to normal functions of apps, *Full Internet Access*, participants had showed ambivalence, and the recommendation acceptance rate varied around 70 percent. We also can find the permissions, such as *Modify/Delete contents*, did not obtain high acceptance. It is about crossing between privacy and usability. Users will not have an unambiguous preference to a permission if most apps request it for functioning. For instance, according to an investigation involved 34,369 apps, more than 85 percent apps request *Modify/Delete contents* permission [24] so that users' ambivalence makes this permission have low acceptance in our method. One possible reason is that such permissions are at the crossing point between privacy and usability, thus is hard to make right recommendations. More specifically, in general these permissions are requested by the apps who have relevant functions so users may think it is fine to allow such apps to grant these permissions. However, users may still have concerns about these permissions since they indeed can lead to privacy risks. Thus, users have ambivalent attitude toward these permissions and the corresponding accuracy is relatively low compared with others accordingly. Such an observation actually reflects the decision trade-offs between privacy concerns and desires to use one specific app.

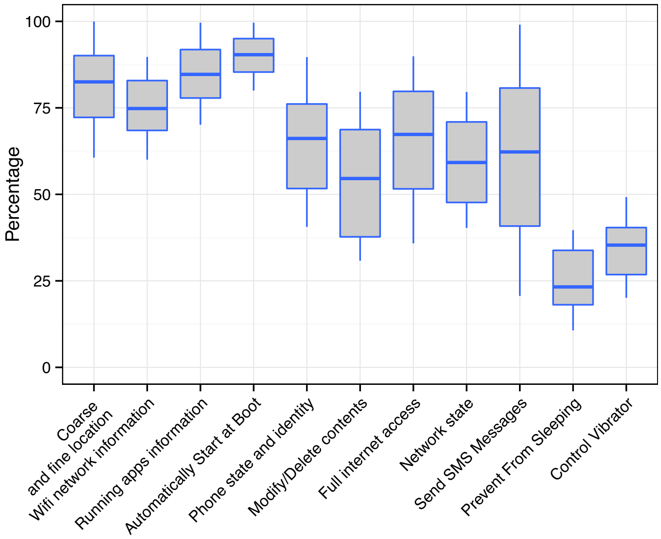


Fig. 8. The percentage of apps that users take the recommendations of each data permission.

To evaluate our other objective, i.e., improving awareness of privacy preference, we depicted the results according to the feedback in Fig. 9. From the graph, we can see that participants have a better comprehension 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 wifi network is permeating our life in every aspect inevitably and people take it as a kind of routine. Thus, participants did not feel remarkable improvement of awareness of wifi network information.

After the experiments ended, we presented participants with a questionnaire about how they feel regarding their privacy of their mobile apps in the smartphones. Twenty-three participants responded. Many participants noted that PriWe increased their awareness of privacy risks. Some comments from the participants, which are retouched for better reading, are listed as follows,

"At the beginning, I didn't care about my privacy at all when I use applications, but when I saw the data usage in my screen, I realized that some sensitive information may be disclosed."

"PriWe looks interesting and I also want to use it to check my privacy before using some apps."

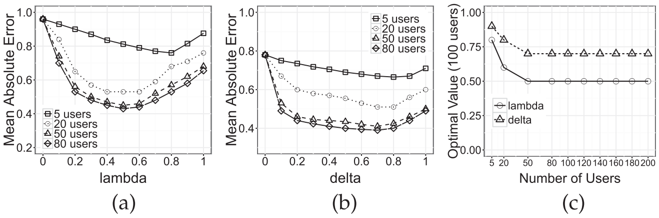


Fig. 10. Parameters estimation of the recommendation algorithm. (a) the impact of lambda (b) the impact of delta (c) the impact of size of participants

5.3 Parameters Estimation

There are two parameters, λ and δ, in the Eq. (17). Since the concrete value of both parameters are calculated from data from real world deployments, it is important to know their stability and scalability. By stability, we mean the optimal value of a parameter will not change greatly with the concrete data in different sets. By scalability, we mean the optimal value of a parameter will not change greatly with the dataset size. The optimal value here is defined as the value that will lead to minimized *Mean Absolute Error* (MAE) [27] shown in Eq. (19) where L denotes the total number of predicted permission settings. The basic idea of MAE is to calculate the average absolute deviation of predictions to the ground truth data. In our study, several sub-datasets with different size and content were generated randomly from original testing dataset, and we computed the mean absolute error of our recommendation results to the actual selections of the participants for each sub-dataset under different parameter values. In order to remove impact of the other parameter, we first tested λ by setting δ to zero, and later nail down λ to test δ. The results are shown in Fig. (10)

$$MAE = \frac{\sum_{x,y,z} |r_{x,y,z} - \hat{r}_{x,y,z}|}{L}. \tag{19}$$

Fig. 10a presents MAE of recommendation results by varying λ from zero to one under multiple datasets with different sizes (i.e., with 5, 20, 50 and 80 participants respectively). It shows that our recommendation algorithm can achieve minimal mean deviation error when λ falls in range between 0.4 and 0.6, and further calculations will output 0.5 as the optimal value for λ. The data in Fig. 10a also shows that the optimal value of λ is fixed around 0.5 under different dataset and different set sizes, thus such an optimal value is stable and can scale well with different datasets.

The results of parameter δ are given in Fig. 10b which show that the optimal value of δ is also pretty stable and scales well. More specifically, its optimal value is 0.7, because under all cases, our recommendation algorithm will always achieve minimal MAE when δ equals to 0.7, even though the datasets contents and sizes have changed dramatically.

To better understand the scalability of both parameters, i.e., how the number of participants would impact the optimal value of λ and δ, some additional experiments were done and the results are given in Fig. 10c. It showed that: when the dataset size is small, the optimal values of both parameters will change greatly. However, when the dataset size is larger than 50, their optimal values become very stable and will stick to 0.5 and 0.7 respectively. This means that we need only to learn the optimal parameter values once with a big initial dataset, and such learned optimal values can be effective for a later real world deployment in large scale.

6 Related Work

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

6.1 Security Protection for Mobile Apps

According to recent systematically research, several vulnerabilities have existed in Android apps. Their presence is even in some extremely popular apps [28]. 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 [29], [30], [31], [32], [33], [34], [35]. Permission analysis is a telling method to detect sensitive information potential leakage [32]. Some static analysis tools have also been developed to automatically detect attempts to load external code using static analysis techniques [29], [36]. Access control provides a different perspective of security and privacy detection and protection in Android system [37], [38]. FlaskDroid [30] privodes mandatory access control on Android's middleware and kernel layers to prevent information disclosure. AppIntent [33] provides a framework which tries to control data transmission to prevent Android applications from stealing sensitive data, meanwhile identify if transmission is from users' intention. TaintDroid [31] is a notable dynamic taint tacking and analysis system, which involves some aforementioned methods to simultaneously track multiple sources of sensitive data. All these works put more 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 [39], [40].

MockDroid [34] can provide artificial data instead of real one to the apps such that they can still function. In this case, there is, actually, no risk for users because the data is fake. However, due to the same reason, apps cannot provide competent services to users. Our work also can produce fake data. The key differentiator of PriWe as compared to prior work is that PriWe produces fake data based on users' expectation of privacy of apps. That means, PriWe only provides fake data when it is sensitive to the user to balance the trade-off between privacy and services.

Besides Android operating system, some research works also make efforts on iOS. PMP is in use for over nine months by 90,621 real users and 225,685 unique apps were reviewed. Based on the crowdsourcing, PMP can recommend users the decisions for their permission settings in iOS [41]. However, PMP is initialized by crowdsourced experts' opinions to generate decisions. In our work, we choose normal users' preferences to boot the system since we believe privacy should be driven by individual attitude and preference.

6.2 System Permissions of Android Apps

Android provides security to users through a permission mechanism [21]. 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 can they 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 [17], [42]. Thus, a mechanism, called permissions removal, has been proposed to mitigate the privacy leak in Android smartphone [19]. 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 [16], [43]. 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 developing habits [20]. 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 [44]. According to the related work, users have an equivocal understanding of mobile apps' permissions, and most of them put little effort to canvass them. Therefore, it is crucial to understand users' privacy concerns and help them make decisions.

6.3 Understanding Privacy and Decision Making System

Before a review of work about understanding privacy and decision making system, we recall the discussion in Section 2. 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 [3]. Therefore, privacy of Android apps should emphasize that users have adequate awareness and understanding to their personal, even sensitive information. According to a recent survey [45], 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 [46], is proposed to capture people's expectations of privacy. Appprofiler [47] is an approach to provide users with knowledge for decision-making about Android application through analyzing privacy-related behaviors of apps and users' opinions.

Past work about understanding privacy of smartphone users indeed 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. In our work, PriWe captures the information through an Android app and learns users' privacy concerns and preferences based on the collected data. Hence, PriWe makes recommendations to users based on their expectations.

7 Discussion

In this section, we discuss some possible limitations of work, which may be argued.

First, we discuss the conception of privacy, since it is the foundation of our work. Nowadays, there is still no universally agreed-on definition of privacy in either research community or industry. We prefer to follow some prevalent interpretations. For example, "privacy is private life, habits, act, relations and the right to be alone [2]" and "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 [3]". Like we pointed in Section 2, these two acknowledged definitions both emphasized that privacy to people should be an ability to express themselves selectively. Therefore, we believe the privacy in the mobile is also based on users' expectations and decisions, especially in the current information age. It is very difficult to protect every single piece of data of each user, so the users have to express their expectations of privacy and we can help them to mitigate the corresponding privacy risk accordingly.

Second, we initialize the recommendation mechanism according to the collected users' privacy permission settings rather than the experts' opinions. This is something about our design philosophy, which is mentioned in Section 2. We consider there is no right answer for the people who want to set their privacy permission settings. In our case, we get some privacy permission settings from the participants during the experiments. Furthermore, the people are also allowed to make their choice on the privacy permission settings when they start to use our services. These two sources will be regarded as the initial dataset for generating the recommendations for each user.

Third, we discuss the parameters in the recommendation approach in Section 5.3. We determined the parameters according to the MAE of recommendations. Also, according to our illustration, the number of participants also influences the performance of recommendation algorithms. The research issue about participant selection for generating recommendation algorithm is proposed, which is out of scope of this article and will be the future work.

Fourth, there are more than 400 participants involved in our work to help us conduct the experiments and improve our research. We admitted that the more people participate, the better the results will be. However, we cannot recruit as many participants as possible due to the time and resource limitation, even though, we try our best to get more users involved. All the information about the participants are shown in Table 2. We avoid the statistical bias of the population, which can make our results more convincing.

Finally, there are two kinds of experiments to evaluate PriWe as shown in Section 5. One is based on Amazon Mechanical Turks, the other one is based on the real deployment. Both of them are based on the real users in the world. In the Amazon Mechanical Turks, we can get more participants in easily, which is significant to our work. In the real deployment, people will use our app and provide more feedback to us since we can have the face-to-face survey, which also can help us to improve our work. That is the reason why we conduct two sorts of evaluation.

8 Conclusion

In this paper, we proposed PriWe, a system aimed to understanding users' expectations of privacy and making recommendations about their privacy settings of installed mobile apps accordingly. Based on the feedback of 422 participants to our survey, the recommendation made by PriWe can achieve around 79 percent accuracy for all the participants and achieve about 90 percent accuracy for the people in information privacy and security area. According to the feedback of 78 users from the real world, PriWe can make proper recommendations which can meet participants' privacy expectations and are mostly accepted by users, thereby help them to mitigate privacy disclosure in smartphone apps.

Acknowledgments

This work is supported by HK GRF Grant No. 24207815, NSFC/RGC Joint Research Scheme N_PolyU519/12, NSFC Key Project No. 61332004, NSFC Grant No. 61572415 and 61562005.

References

[1] Android M permissions: Best practices for developers. (2016). [Online]. Available: https://developer.android.com/preview/features/ runtime-permissions.html

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

[3] A. F. Westin, "Privacy and freedom," *Washington Lee Law Rev.*, vol. 25, no. 1, 1968, Art. no. 166.

[4] H. Nissenbaum, "Privacy as contextual integrity," *Washington Law Rev.*, vol. 79, 2004, Art. no. 119.

[5] R. Liu, J. Cao, S. VanSyckel, and W. Gao, "Prime: Human-centric privacy measurement based on user preferences towards data sharing in mobile participatory sensing systems," in *Proc. 14th IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 14–19, 2016, pp. 1–8.

[6] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1268–1282, Jun. 2014.

[7] A. Liu, Q. Zhang, Z. Li, Y.-J. Choi, J. Li, and N. Komuro, "A green and reliable communication modeling for industrial internet of things," *Comput. Elect. Eng.*, (2016, Sept.). [Online]. Available: http://dx.doi.org/10.1016/j.compeleceng.2016.09.005

[8] H. Dai, G. Chen, C. Wang, S. Wang, X. Wu, and F. Wu, "Quality of energy provisioning for wireless power transfer," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 2, pp. 527–537, Feb. 2015.

[9] X. Liu, "A transmission scheme for wireless sensor networks using ant colony optimization with unconventional characteristics," *IEEE Commun. Lett.*, vol. 18, no. 7, pp. 1214–1217, Jul. 2014.

[10] Q. Ismail, T. Ahmed, A. Kapadia, and M. K. Reiter, "Crowdsourced exploration of security configurations," in *Proc. 33rd Annu. ACM Conf. Human Factors Comput. Syst.*, 2015, pp. 467–476.

[11] R. Liu, J. Cao, L. Yang, and K. Zhang, "PriWe: Recommendation for privacy settings of mobile apps based on crowdsourced users' expectations," in *Proc. 4th IEEE Int. Conf. Mobile Services*, Jun. 27–Jul. 2, 2015, pp. 150–157.

[12] J. Lee, M. Sun, and G. Lebanon, "A comparative study of collaborative filtering algorithms," *CoRR*, 2012. [Online]. Available: http://arxiv.org/abs/1205.3193

[13] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 1999, pp. 230–237.

[14] 12 most abused android App permissions, 2013. [Online]. Available: http://about-threats.trendmicro.com/us/library/image-gallery/12-most-abused-android-app-permissions

[15] 92% of top 500 android apps carry security or privacy risk, 2014. [Online]. Available: http://www.infosecurity-magazine.com/news/92-of-top-500-android-apps-carry-security-or/

[16] C. Orthacker, et al., "Android security permissions-can we trust them?" in *Security and Privacy in Mobile Information and Communication Systems*. Berlin, Germany: Springer, 2012, pp. 40–51.

[17] 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*. Berlin, Germany: Springer, 2012, pp. 68–79.

[18] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *Proc. Symp. Usable Privacy Secur.*, 2012, Art. no. 3.

[19] Q. Do, B. Martini, and K.-K. R. Choo, "Enhancing user privacy on android mobile devices via permissions removal," in *Proc. 47th Hawaii Int. Conf. Syst. Sci.*, 2014, pp. 5070–5079.

[20] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2011, pp. 627–638.

[21] Android system permissions, 2014. [Online]. Available: http://developer.android.com/guide/topics/security/permissions.html

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

[23] Xposed module repository. (2016). [Online]. Available: http://repo.xposed.info/

[24] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, "An empirical study on android for saving non-shared data on public storage," in *Proc. 30th IFIP Int. Inf. Secur. Conf.*, 2015, pp. 542–556.

[25] G. McLachlan, K.-A. Do, and C. Ambroise, *Analyzing Microarray Gene Expression Data*, vol. 422. Hoboken, NJ, USA: Wiley, 2005.

[26] The statistics portal, 2014. [Online]. Available: 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

[27] G.-R. Xue, et al., "Scalable collaborative filtering using cluster-based smoothing," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2005, pp. 114–121.

[28] Y. Zhou and X. Jiang, "Detecting passive content leaks and pollution in android applications," in *Proc. 20th Annu. Netw. Distrib. Syst. Secur. Symp.*, 2013.

[29] S. Poeplau, Y. Fratantonio, A. Bianchi, C. Kruegel, and G. Vigna, "Execute this! analyzing unsafe and malicious dynamic code loading in android applications," in *Proc. Annu. Netw. Distrib. Syst. Secur. Symp.*, 2014, vol. 14, pp. 23–26.

[30] S. Bugiel, S. Heuser, and A.-R. Sadeghi, "Flexible and fine-grained mandatory access control on android for diverse security and privacy policies," in *Proc. 22nd USENIX Secur. Symp.*, 2013, pp. 131–146.

[31] W. Enck, et al., "TaintDroid: An information flow tracking system for real-time privacy monitoring on smartphones," *Commun. ACM*, vol. 57, no. 3, pp. 99–106, 2014.

[32] 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 *Proc. ACM Conf. Comput. Commun. Secur.*, 2010, pp. 73–84.

[33] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "Appintent: Analyzing sensitive data transmission in android for privacy leakage detection," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2013, pp. 1043–1054.

[34] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, "MockDroid: Trading privacy for application functionality on smartphones," in *Proc. Workshop Mobile Comput. Syst. Appl.*, 2011, pp. 49–54.

[35] A. Hossain, A.-A. Hossain, S.-J. Jang, and J.-W. Chang, "Privacy-aware cloaking technique in location-based services," in *Proc. IEEE Int. Conf. Mobile Services*, 2012, pp. 9–16.

[36] P. Yang, X. Xie, I. Ray, and S. Lu, "Satisfiability analysis of workflows with control-flow patterns and authorization constraints," *IEEE Trans. Services Comput.*, vol. 7, no. 2, pp. 237–251, Apr.–Jun. 2014.

[37] W. She, I.-L. Yen, B. Thuraisingham, and E. Bertino, "Security-aware service composition with fine-grained information flow control," *IEEE Trans. Services Comput.*, vol. 6, no. 3, pp. 330–343, Jul.–Sep. 2013.

[38] M. Shehab and F. Mohsen, "Towards enhancing the security of OAuth implementations in smart phones," in *Proc. IEEE Int. Conf. Mobile Services*, 2014, pp. 39–46.

[39] R. Liu, J. Cao, S. VanSyckel, and W. Gao, "PriMe: Human-centric privacy measurement based on user preferences towards data sharing in mobile participatory sensing systems," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, Mar. 2016, pp. 1–8.

[40] R. Liu, J. Cao, K. Zhang, W. Gao, J. Liang, and L. Yang, "When privacy meets usability: Unobtrusive privacy permission recommendation system for mobile apps based on crowdsourcing," *IEEE Trans. Services Comput.*, vol. PP, no. 99, p. 1, 2016, Doi: 10.1109/TSC.2016.2605089.

[41] Y. Agarwal and M. Hall, "ProtectMyPrivacy: Detecting and mitigating privacy leaks on ios devices using crowdsourcing," in *Proc. Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 97–110.

[42] A. Kliem, M. Hovestadt, and O. Kao, "Security and communication architecture for networked medical devices in mobility-aware ehealth environments," in *Proc. IEEE Int. Conf. Mobile Services*, 2012, pp. 112–114.

[43] R.-H. Hwang, Y.-L. Hsueh, and H.-W. Chung, "A novel time-obfuscated algorithm for trajectory privacy protection," *IEEE Trans. Services Comput.*, vol. 7, no. 2, pp. 126–139, Apr.–Jun. 2014.

[44] 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 Trans. Softw. Eng.*, vol. 40, no. 6, pp. 617–632, Jun. 2014.

[45] Z. Benenson, F. Gassmann, and L. Reinfelder, "Android and iOS users' differences concerning security and privacy," in *Proc. ACM CHI Conf. Human Factors Comput. Syst. Extended Abstracts*, 2013, pp. 817–822.

[46] 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 *Proc. ACM Conf. Ubiquitous Comput.*, 2012, pp. 501–510.

[47] S. Rosen, Z. Qian, and Z. M. Mao, "AppProfiler: A flexible method of exposing privacy-related behavior in android applications to end users," in *Proc. ACM Conf. Data Appl. Secur. Privacy*, 2013, pp. 221–232.

Rui Liu received the BEng degree from Northeastern University, China, and the MPhil degree from Hong Kong Polytechnic University. He is currently a research assistant in the Department of Information Engineering, Chinese University of Hong Kong. His research interests include ubiquitous computing, mobile computing, and crowdsourcing. He is a member of the IEEE.

Junbin Liang received the BSc and MSc degrees in computer science from Guangxi University, in 2000 and 2005, respectively, and the PhD degree from Central South University, China in 2010. He is currently a professor with Guangxi University. His research interests include mobile ad hoc networks, wireless sensor networks and real-time programming language design.

Jiannong Cao received the BSc degree in computer science from Nanjing University, China, and the MSc and PhD degrees in computer science from Washington State University. He is currently a chair professor and the head of the Department of Computing, Hong Kong Polytechnic University. His research interests include parallel and distributed computing, computer networks, mobile and pervasive computing, fault tolerance, and middleware. He co-authored four books, coedited nine books, and published more than 300 technical papers in major international journals and conference proceedings. He is a fellow of the IEEE, a member of the ACM, and a senior member of the China Computer Federation.

Kehuang Zhang received the PhD degree in informatics from Indiana University at Bloomington, in 2012. He is an assistant professor in the Information Engineering Department, Chinese University of Hong Kong. His current research focuses on system and software security, including mobile computing security, cloud computing, and embedded system security. He is a member of the IEEE.

Wenyu Gao received the BSc degree (double major in statistics and finance) from the University of Hong Kong, in 2013 and the MA degree in statistics from Columbia University, in 2014. She is currently working toward the PhD degree in the Statistics Department, Virginia Polytechnic Institute, and State University. Her research interests include machine learning, human-computer interaction, and bayesian statistics.

Lei Yang received the BSc degree from Wuhan University, in 2007, the MSc degree from the Institute of Computing Technology, Chinese Academy of Sciences, in 2010, and the PhD degree from the Department of Computing, Hong Kong Polytechnic University, in 2014. He is currently a post-doctoral fellow in the Department of Computing, Hong Kong Polytechnic University. His research interests include mobile cloud computing, big data, and internet of things. He is a member of the IEEE.

Ruiyun Yu received the bachelor's degree in mechanical engineering from the Northeastern University, China, in 1997, and the MS and PhD degrees in computer science, in 2004 and 2009, respectively. He is currently a professor and vice dean of the Software College, Northeastern University, China. From September 2006 to October 2007, he worked as a visiting scholar in the Center for Research in Wireless Mobility and Networking, University of Texas at Arlington. His research interests include participatory sensing systems, wireless sensor networks, mobile and pervasive computing, and virtual simulation technology. He is a member of the IEEE.