Hindawi Publishing Corporation
Mobile Information Systems
Volume 2016, Article ID 6804379, 10 pages
http://dx.doi.org/10.1155/2016/6804379

Research Article

How Dangerous Are Your Smartphones? App Usage Recommendation with Privacy Preserving

Konglin Zhu,[1] Xiaoman He,[1] Bin Xiang,[1] Lin Zhang,[1] and Achille Pattavina[2]

[1]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China*
[2]*Dipartimento di Elettronica e Informazione, Politecnico di Milano, 20133 Milano, Italy*

Correspondence should be addressed to Konglin Zhu; klzhu@bupt.edu.cn

Received 30 December 2015; Revised 31 March 2016; Accepted 24 May 2016

Academic Editor: Mea Wang

With the rapid proliferation of mobile devices, explosive mobile applications (apps) are developed in the past few years. However, the functions of mobile apps are varied and the designs of them are not well understood by end users, especially the activities and functions related to user privacy. Therefore, understanding how much danger of mobile apps with respect to privacy violation to mobile users is becomes a critical issue when people use mobile devices. In this paper, we evaluate the mobile app privacy violation of mobile users by computing the danger coefficient. In order to help people reduce the privacy leakage, we combine both the user preference to mobile apps and the privacy risk of apps and propose a mobile app usage recommendation method named AppURank to recommend the secure apps with the same function as the "dangerous" one for people use. The evaluation results show that our recommendation can reduce the privacy leakage by 50%.

1. Introduction

The rapid growth of mobile devices has been leading to the prosperity of mobile applications (apps). For example, as of the end of 2014, the number of apps on Play Store has been over 1.4 million and over 1.2 million on Apple Store. This number is still growing dramatically with the proliferation of mobile devices. Mobile users download these mobile apps and use them on their mobile devices to satisfy different purposes. However, the functions of mobile apps are varied and the designs of them are not well understood by end users, especially the activities and functions related to user privacy. Indeed, to improve user experience and the functionality of mobile apps, developers start to move their eyes on the personalized service that can be provided by apps. They develop new functional apps or enhance the capability of apps by digging into the personal information, such as location information, contacts, camera, messaging, and even calling service. However, when users launch apps on their mobile devices, they may fall into danger as some unknown activities or functions might cause privacy issue.

Although app stores (e.g., Google Play) may remove those apps with malfunctions or low quality periodically, many well developed apps with privacy violations are not perceived by the stores. In other words, app stores release the right to end users to let them decide whether to install the apps or not. In such case, most end users download and install those apps by neglecting the privacy warning. Even when some users do notice the privacy issue, as the functionality and preference of mobile apps, they still install them into the mobile device. Once the apps are installed, users' privacy information will be leaked when they launch those apps.

As mobile apps serve for different functionalities, different types of privacy information may be leaked by users for launching different types of mobile apps. For instance, a location-based service (LBS) needs to collect user real time location information from users. It may refer to home and workplace where users may not expect to be exposed. A social-aware service needs to extract contacts from users, which violates many users' personal life as well, not to mention the information collected for function-unrelated purpose. In fact, it is reported that users have growing

concern about their privacy while using mobile apps. A recent survey from IDG news reveals that over 30% of mobile users prefer to uninstall those apps after learning the personal information they collected. Unfortunately, users do not know how much of the personal information has been collected and how much danger of mobile apps with respect to privacy violation to mobile users. Therefore, it is crucial to understand how dangerous are installed apps on mobile devices as well as how much privacy risk is taken by launching those apps. It will also be beneficial for mobile users to know how to reduce the privacy leakage by using those apps with less privacy concern and meanwhile maintain the quality of experience.

Thus far, majority of mobile app recommendation approaches have been developed based on the popularity of apps while neglecting the privacy issues existing in mobile apps [1–3]. Several privacy-concerned app detection and recommendation mechanisms are proposed to discover the malfunction of mobile apps. They either focus on the service provider side to let app stores recommend those apps with less privacy concern without considering the user personalization [4] or target the developer side to investigate the inside of apps, find malware code, and restrict the app access permission [5–9]. However, these actions need the cooperation with either service providers or apps developers, which makes them difficult to be implemented in practice.

In this paper, in order to measure how much privacy a mobile user attempts on the mobile phone and help to recommend apps with less privacy, we propose a privacy evaluation mechanism by analyzing the app usage data. The violation of privacy depends not only on the risks of apps, but also on the user usage pattern. Although some apps violate user privacy heavily, user information cannot leak if the app has not been used. To evaluate the privacy leakage, we define a danger coefficient to quantify the privacy and analyze the privacy violation distribution of mobile users. To reduce the privacy leakage from mobile devices, on one hand, we need to understand how much privacy a mobile app can expose. On the other hand, we need to investigate the apps usage of different users. Therefore, we combine both the user preference to mobile apps and the privacy risk of mobile apps for apps recommendation. To formulate the user preference to mobile apps, we apply the distribution of common preference probabilistic method, which can enrich the context of personalized preference. To understand the privacy violation of mobile apps, we measure the privacy access permissions of mobile apps. Finally, we seek a balance between the user preference to apps and the privacy violation of apps to propose an app usage recommendation method named AppURank. According to the functionality of mobile apps, we classify them into groups by the topic model. The proposed approach recommends the same functional apps with lighter privacy concern and high user preference. We evaluate our mobile app recommendation approach with extensive experiments. The results show that our proposed recommendation method can halve the danger of mobile devices and meanwhile maintain the same level of user preference.

The contributions of the paper are summarized as follows:

(i) We carry out a mechanism to evaluate the extent of privacy leakage when people use their mobile devices. We define a danger coefficient to measure the privacy violation of apps from usage perspective and analyze the privacy violation distribution of mobile users.

(ii) We propose an app usage recommendation approach for end users, named AppURank, by combining user preference, privacy risks, and functionality of apps. The proposed method is to recommend people with preferred apps but with less privacy violation.

(iii) We evaluate the app recommendation method on our collected data. It shows that the method can reduce the danger of mobile devices to half and meanwhile maintain the same level of usage preference.

The rest of the paper is structured as follows: Section 2 reviews the related literature. Section 3 describes the problem, introduces the definition of danger coefficient of mobile users, and provides a recommendation on the app usage that can minimize the risk of privacy leakage. Section 4 shows the experimental results. Conclusions are finally given in Section 5.

2. Related Literature

In this section, we review the state of the art for the privacy leakage evaluation of mobile apps and mobile app recommendation approaches.

The privacy issue of mobile apps has been studied for many years. One group of previous studies regarding privacy issue of mobile apps concerns the risk analysis of mobile apps. For instance, Au et al. [6] surveyed the permission systems of smartphone operating systems from the amount of controls, the information released to users, and the levels of interactivity from users. Felt et al. [5] focused on the permission request of various mobile apps to determine whether Android developers follow least privilege with their permission requests. They further built Stowaway to detect overprivilege in Android apps. Enck et al. [7] proposed TaintDroid, which provided real time analysis of mobile apps on the monitoring of their data access by leveraging Android's virtualized execution environment. In contrast, majority of mobile app privacy studies are proposed for privacy violation or mobile app malfunction detection. To deal with the information stealing of mobile apps, Zhou et al. [8] carried out the TISSA system, which can empower users to flexibly control the accessibility of mobile apps to personal information. Enck et al. [9] exploited a rule-based certification model named Kirin to perform lightweight certification of mobile apps at installing time to reduce the privacy issue caused by mobile apps.

However, these mechanisms need investigation on the code installed in mobile apps in order to identify the privacy violation, which is difficult for all apps installed on mobile devices. Meanwhile, people do not like apps scanning their mobile devices all the time. In this paper, we propose the

privacy preserving method by recommendation approach, which can avoid the installation of risky apps.

The majority of mobile app recommendation methods consider the popularity or user preference as factors for the decision making. For instance, AppJoy [1] made personalized mobile app recommendation by analyzing how users actually use their installed apps. They applied collaborative filtering algorithm for individual recommendation. Yu et al. [2] and Zhu et al. [3] considered the user context for mobile app recommendation and used Latent Dirichlet Allocation (LDA) topic model to describe the problem of mobile apps recommendation. Few papers focus on the mobile apps based on privacy issue. Peng et al. [10] proposed a risk ranking method of Android apps using probabilistic generative model to tell users the privacy risk of mobile apps before installation. The other one by Zhu et al. [4] used the modern portfolio theory to recommend mobile apps from the perspective of app store by considering the awareness of security and privacy. Our paper tries to recommend mobile apps from personalized user-app usage perspective to avoid privacy violation. There are some literatures focusing on permission settings recommendation to preserve user privacy. Lin et al. [11] proposed to provide reasonable default settings to help users configure their privacy settings by identifying distinct privacy profiles. Liu et al. [12] proposed and implemented PriWe, which leveraged the crowd sourced permission settings to understand users' privacy expectation and provides app specific recommendations to mitigate information leakage.

We use the similar methods as [10] to obtain the privacy risk of mobile apps. However, different from the state of the art [4, 10], we not only consider the privacy risk of mobile apps in general, but also combine the function relativity of mobile apps to the permissions and people usage patterns of mobile apps to evaluate the privacy danger of mobile apps to users. We then propose a mobile app recommendation method by considering both user preferences and mobile app privacy.

3. Problem Formation

3.1. Preliminaries. When an app is installed or launched on mobile device, it always asks the permission to access certain information. The permission means the capability users grant to mobile apps so that mobile apps can access certain part of mobile users' information. These permissions are associated with mobile apps to either help the mobile apps to achieve some functions (e.g., localization) or fulfill the mobile apps to collect user data. The permissions requested by a mobile app are independent of each other. When mobile app is installed or launched on the smartphone, the users have the right to make decision for the app to access the permission of information. In fact, the information those apps intend to access may refer to the sensitive personal private data, such as location information and the control of hardware (e.g., camera). For instance, Table 1 illustrates the access permission list requested by a version of Google search app in an app store. It can be seen that some information requested by the app permissions, such as network connection and storage, is necessary for the function of the app (i.e., information searching), referred here to as "function-related" for the app. At the same time some other information requested by the app permissions, such as phone state, personal data, payment service, and system tools, is not highly correlated with the function of the app, referred here to as "function-unrelated" for the app. As a matter of fact, for different types of permissions, the degree to which they violate the privacy is different. For instance, as shown in Table 1, the payment service permission that allows running the operation for payment is more severe than the permission of network connection. There are two reasons for such a judgement. First, the payment service permission may cause the economic loss, which is more vital than the network connection. Second, the payment service is function-unrelated while network connection is function-related for searching function. Therefore, all the information involved in an app can be categorized into several tiers according to its degree of privacy violation and its relativity to app function: (a) "normal permission," which does not involve sensitive information of mobile users, such as network connection and storage; (b) "severe permission," meaning the information is severely related to user privacy, such as personal information, location, and payment service; and (c) "system permissions," related to the control of hardware and system, such as the access of hardware and setup of system level configuration. Therefore, combining the function and the extent of privacy violation, six different types of information permission are considered: (1) function-related with normal permission, (2) function-related with severe permission, (3) function-related with system permission, (4) function-unrelated with normal permission, (5) function-unrelated with severe permission, and (6) function-unrelated with system permission.

Table 1: Access permission of a search engine app.

Permission	Description
Network connection	Allow the app to access the Internet
Storage	Allow accessing external SD card
Phone state	Allow accessing phone information
Personal information	Allow accessing contact information, messages, emails, and so forth
Location	Allow accessing the geographical information instantly
Hardware	Allow accessing camera, audio, and recorder
Payment service	Allow running the operation for payment
System tool	Allow setting up the display

Although apps violate user privacy by permissions of accessing user information, mobile users still install and launch different types of mobile apps on their smartphones according to their preferences. In order to recommend apps by considering both user privacy preserving and user preference, in this paper, we discuss three issues: (1) how to measure the privacy risk of an app and the privacy violation to mobile users by launching mobile apps, (2) how to determine

user preference to mobile apps, and (3) how to balance the privacy violation to mobile users and the user-app preference to meet the requirement of users.

In the following, we will define the danger coefficient to quantify the privacy violations of permissions to mobile users and also address the above-mentioned three issues to recommend mobile apps from usage perspective.

3.2. Danger Coefficient. Generally speaking, the privacy information is normally leaked when people launch an app with privacy permission. We introduce here a new parameter, called *danger coefficient* (DC), capable of expressing the leakage of privacy when users run apps on their mobile device.

To evaluate the DC of each user, two factors need to be determined. The first one is the privacy risk of each app, which is reflected by the permissions that the app asks from users. We measure the privacy risk of permissions requested by the app and consider it as one factor for evaluating the app's DC. The second one is the app usage pattern by the user. In what pattern the mobile app is used indicates the probability that the privacy information disclosed by the app will be leaked, which is considered as the other factor for evaluating app's DC.

We now address the problem by characterizing and quantify the privacy risk of permissions and then the privacy risk of an app. Permissions can be classified into three classes, which are normal permissions, critical permissions, and system permissions as we discussed in the previous section. Requesting a more critical permission increases risk more than requesting a less critical one. For the quantification, we leverage the probabilistic approach proposed in [10] to evaluate the risk of each of the categories of permissions.

For the generic app a_i ($i = 1, \ldots, M$), the permissions that will be accessed are denoted by the set $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,N}\}$, where N is the total number of permissions. The generic variable $p_{i,j}$ is binary and assumes value 0 or 1 if permission j is not or is present in app a_i. For mobile apps, different types of permissions may correlate with each other. For instance, permissions related to network (including Internet access, checking WiFi state, checking network status, changing WiFi status, and changing network connection) are mutually correlated. However, such dependence introduces sophisticated analysis to conduct privacy risk evaluation. In contrast, the study [10] discovers that assuming the independence of different permissions can still perform well for the overall privacy risk evaluation but it is more simplistic for the analysis compared with dependence situations. Furthermore, the assumption of independence of different permissions allows a monotonic model, which allows the consideration of each individual permission. Besides, it can also help to differentiate different classes of permissions.

Therefore, if we use $f(P_i)$ to indicate the privacy risk factor for app i, P_i is generated by N independent Bernoulli random variables and is given by

$$f\left(P_i\right) = \prod_{j=1}^{N} r_j^{p_{i,j}} \left(1 - r_j\right)^{1-p_{i,j}}, \tag{1}$$

where r_j is the probability that permission j ($j = 1, \ldots, N$) is accessed by an app.

Following [10], r_j is obtained using a Beta($r_i \mid a_0, b_0$) function. That is,

$$r_j = \frac{\sum_{i=1}^{M} p_{i,j} + a_0}{M + a_0 + b_0}, \tag{2}$$

where M is the total number of apps used for evaluation. In this paper, the value of M is set to 900 as the dataset contains 900 mobile apps. To suggest the several privacy violation risks, we set $a_0 = 1$, $b_0 = 2M$ with less penalty effect for critical permissions. For normal permissions, we set $a_0 = 1$, $b_0 = M1$, which is normal distribution suggesting the less effect of the privacy risk, $a_0 = 1$, $b_0 = M$ with less penalty effect for critical permissions, for normal permissions the value $a_0 = 1$, $b_0 = 1$, which is normal distribution suggesting the less effect of the privacy risk. With this method, different types of permission privacy risks can be distinguished.

Furthermore, apps may request both function-related and function-unrelated permissions. For example, an app providing map service needs location information as its function-related permission, whereas if an app serves as chatting service, requesting calling permission will very likely be considered as function-unrelated permission. If an app requests a function-related permission, the privacy violation is considered much weaker than that of function-unrelated permission request. To show the difference, we assign different weights for the privacy risk and define the weight factor $\omega_{i,j}$ of permission j for app a_i. The weights of the function-related permissions should be less than that of the function-unrelated permissions, as the function-related permissions are about to enable functions and services, whereas the function-unrelated permissions intend to collect user privacy information. For instance, we take an empirical value that the weight for function-related permissions is 0.5 and for function-unrelated permissions is 1. Then the overall risk factor $f(a_i)$ of app a_i that takes into account the weight of the different risks is given by

$$f\left(a_i\right) = f\left(P_i\right) * \prod_{j=1}^{N} \left(\omega_{i,j}\right). \tag{3}$$

Considering the app privacy risk is monotonically decreasing with respect to the probability of using granted permissions, which means removing a permission always reduces the risk value of an app, for privacy risk calculation of an app, we use the following function [10]:

$$R\left(a_i\right) = -\ln\left[f\left(a_i\right)\right]. \tag{4}$$

As far as the app usage pattern is concerned, a user with longer time duration of using an app will have more chance to access privacy information. Although there are some mobile apps that are launched in the background and steal the user privacy information without being invoked by users, they are not measured and counted in our analysis. In this paper, we assume that the probability that a mobile app accesses privacy information is proportional to the time duration in which the app is used. Therefore, we measure the usage pattern by

expressing the fraction of time user m ($m = 1, \ldots, L$) in which uses app a_i; that is, $U_{i,m} = t_{i,m}/T$, where $t_{i,m}$ is the total time usage of app a_i by user m and T is the total observation interval of the system.

The state of the art suggests that interevent times of human behaviors follow Poisson distribution [13]. In this paper, we assume the interusage time follows a Poisson distribution with the parameter equal to $\lambda_{i,m}$ for user m running app a_i. Then $U_{i,m} = 1 - 1/\lambda_{i,m}$.

The DC measures the danger degree of a user using a mobile device. For a user u_m, the danger coefficient can be derived by combining the app privacy risk and the app usage pattern and is expressed as

$$\mathrm{DC}(u_m) = \sum_{i=1}^{M} [R(a_i) * U_{i,m}], \tag{5}$$

where M is the number of apps launched in the device. The larger $\mathrm{DC}(u_m)$ indicates the more chance for privacy violation when user u_m launches apps in the mobile device. For a user with multiple devices, each device can be evaluated by the above procedure for the danger coefficient estimation. We use DC as a metric to evaluate the danger degree of the recommendation algorithms in Section 4.

3.3. User-App Preference. The user-app preference reflects the preference relationship between individuals and apps. As each individual user may not display enough information to fully discover his or her individual preference, we employ an individual user-app preference based on the distribution of common preference, as presented in [3]. Specifically, we firstly investigate the common preference of many users and then represent each user's preference by a distribution of common preference. If the common preferences are presented by z, the conditional probability that a user u_m prefers the category a given the set of all apps A can be represented as

$$P(a \mid A, u_m) = \frac{P(a, A \mid u_m) * P(u_m)}{P(A, u_m)}. \tag{6}$$

For given apps A and user u_m, $P(u_m)$ and $P(A, u_m)$ are constant. Therefore, we have

$$\begin{aligned} P(a \mid A, u_m) &\sim P(a, A \mid u_m) \sim \sum_z P(a, A, z \mid u_m) \\ &\sim \sum_z P(a, A \mid z) P(z \mid u_m), \end{aligned} \tag{7}$$

where the preference of user u_m to the category of apps a is determined by the common preferences of many users (i.e., $P(a, A \mid z)$) and also the user's personal preference conveyed by a distribution of common preferences (i.e., $P(z \mid u_m)$). The calculation of common preferences of other users and the distribution of common preferences of user u_m can be presented by the normalized number of apps launched by user u_m and other users.

The above user-app preference suggests the preference of a user to a category of mobile apps. To identify categories of mobile apps, we first find the categories of each mobile app with its assigned categories. However, such classification is not fine-grained enough for app usage preference recommendation. For example, many mobile apps are associated with offline services, such as online banking apps or social media apps. A Facebook social app cannot be replaced by a Twitter social app. Therefore, for each category of mobile apps, we use a topic model to categorize apps in different coarse grained preference categories into different groups according to their functions and usage context. For this purpose, we employ Latent Dirichlet Allocation (LDA) model [14], in which an app is associated with a word in a document, and each category is a topic. With such topic model, apps are put into different fine-grained preference groups. Specifically, for each user, we extract the context components (i.e., time stamp, and location) from the usage record and consider the set of context components of a user as this user's bag of context components. For the fine-grained preference of each user, the procedure is conducted as follows. It begins with a random assignment of fine-grained preference to each context component. It then iteratively estimates the conditional probability of assigning of the preference to context component and updates the preference of each context component according to the latest calculation. The assignment will converge finally, which means each context component is assigned with a fine-grained preference. Then the user fine-grained preference will be determined by her context component bag. But LDA has some drawbacks to identify the sequence of words and also it is in the topics composition in which the same words appear in the multiple topics. In our case, these drawbacks do not affect the performance of mobile app group identification. For the fine-grained preference classification, only the special words associated with the fine-grained group are used. Moreover, the classification does not involve the sequence of words. Therefore, LDA model is capable of fine-grained classification for mobile apps. If a person launches an app on the mobile device, it is indicated that the person is interested in the functionality provided by the app. If the app is considered with high risk, then another app in the same category should be recommended.

3.4. App Usage Recommendation. As the high risky app may cause more privacy leakage, recommending the user preferred apps with less privacy risk is more desirable. In this section, we will propose a method for app usage recommendation combining required function, user preference, and privacy, named AppURank.

Regarding required functions, we recommend apps which are in the same category as the launched apps to guarantee the function similarity. Then we recommend apps according to preference and privacy. The objective of the app usage recommendation is finding a group of apps $\mathscr{A}$ consisting of a collection of k apps with the corresponding weight α_i and $\sum_{i=1}^{k} \alpha_i = 1$ which display minimal privacy risk and meanwhile satisfy the individual preference and function requirement. The general formulation of the objective can be presented as

$$\max_{\mathscr{A}} \sum_{i=1}^{k} \alpha_i * [P(a \mid A, u_m)] - b * \sum_{i=1}^{k} \alpha_i * R(a_i), \tag{8}$$

where $\alpha_i \in a$ suggests α_i is one mobile app in category a and the value of b depends on the privacy and individual preference requirement of users. For example, if a user considers privacy more important, then the value of b will be larger, while if a user takes user preference more into account, then the value of b will be smaller. Given the vector $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_k]$, the corresponding preference vector and risk vector are represented by P and R. Equation (8) can be rewritten as

$$\max_{\mathscr{A}} \alpha^\top P - b * \alpha^\top R. \quad (9)$$

In this case, P and R are independent, and the selection of α is to rank mobile apps according to user preference and privacy risk according to the calculation of $R(a_i)$ and $P(a \mid A, u_m)$ and then combine the two aspects, as specified by (9). The app with the highest combination value is considered as the most recommended app. Apps with top k values are the top k apps on the recommendation list.

4. Performance Results

In this section, we describe the experimental data and evaluate the privacy risk of mobile apps, danger coefficient of mobile users, and AppURank recommendation method.

4.1. Experimental Data. In this paper, we need two data traces to evaluate the danger coefficient and the proposed recommendation approach.

One data trace contains apps with their requested permissions, and the permissions are marked with function-related and function-unrelated. There are 900 apps and their permissions in the data trace. For each app, we collect the permission information from the privacy description in the store and use a matrix to store them. The rows are apps, and columns are permissions. If an app requests a permission, the element in the matrix will be 1; otherwise, it will be 0. Then we take the function relativity into consideration. We investigate the function relativity in two stages. In the first stage, we crawl the coarse grained categories of mobile apps and identify the permission by their main functions. For example, a location-based social service should need the location information, which means location information is function-related for the location-based social service. In the second stage, we manually identify the other permissions accessed by each of the mobile apps. For instance, an alarm app could require access to microphone to let the user record some memo/voice to be played during timeout. If the permission is requested by malicious functions of an app, it is considered as function-unrelated. For a function-related permission, the value of element is changed to 0.5. For function-unrelated permissions, the value of element is still 1. Then the privacy risk of each app in this data trace can be calculated using the expression of $R(a_i)$ in (4). Figure 1 shows the distribution of apps in terms of the number of requested permissions. It shows that about 20% of apps request less than 10 permissions. The percentage of apps requesting 10 to 20 permissions reaches over 75%, and about 5% of apps request over 20 permissions. On average, each app requests 12 permissions. Figure 2 shows the distribution of apps in terms of different types of permissions. More than 70% of apps request Internet access, phone state, and network state. There are several apps requests for external device format, which highly violate the user privacy and even refer to the security of mobile devices. Regarding function relativity of apps, we draw Figure 3 to show the average number of function-related and function-unrelated permissions of different categories of apps. It shows that function-unrelated permissions are more than function-related permissions in most cases. In the apps of security, car/cab, social, voice, reader, helper, and tools, the function-unrelated permissions are requested twice as much as function-related permissions.

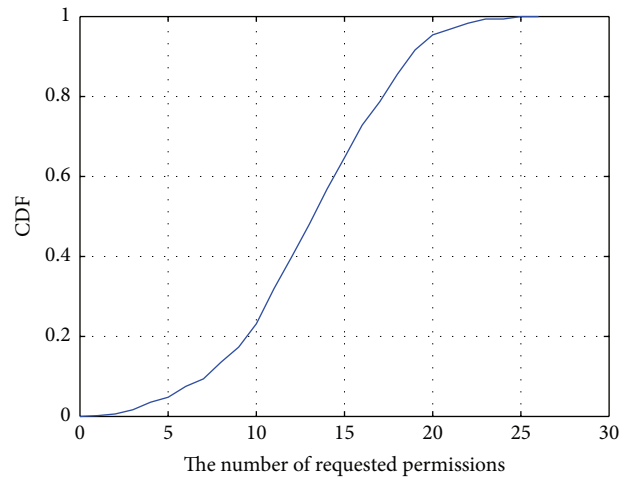


Figure 1: CDF of permission distribution from apps perspective.

Table 2: Characteristics of the dataset.

Characteristics	Value
Users	13,969
Duration	545 days
Network types	2G/3G/4G Wi-Fi
Applications	16,878
Total traffic	13.8 TB
Number of records	Over 103 million

The other data trace is the mobile users and their mobile app usage pattern. In order to obtain such data, we design a mobile app named AntTest (http://www.wandoujia.com/apps/edu.bupt.anttest) on Android platform. Indeed, the AntTest application is developed to measure network speed of mobile device. For such purpose, the data such as app usage pattern of each user is recorded every 5 seconds, which can be exactly applied in this study. We put the AntTest in the app store to provide a way for users to download. So far, the app has been available for more than 600 days since March 2014. The total number of records is over 103 million produced by 13,969 users. The detailed data description is presented in Table 2. The format of one record is presented as $\langle RecordID, IMEI, AppID, Time \rangle$, where *RecordID* is the ID of the record, *IMEI* indicates the user equipment ID, *AppID* is identified by the name and package of the app, and *Time* indicates the time of the record. We anonymize the dataset to conduct the experiment for the privacy concern.

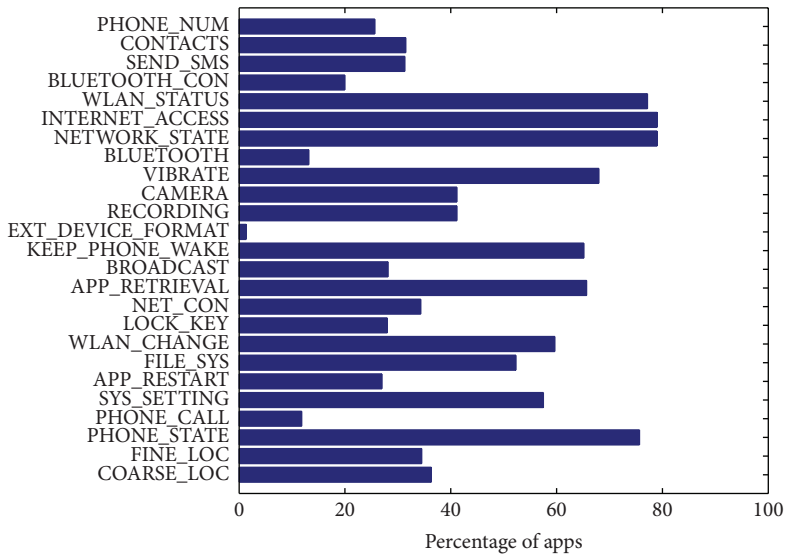


Figure 2: The permission distribution from apps perspective.

4.2. Evaluation of App Usage. In order to evaluate the privacy risk when people launch mobile apps, we measure the app privacy risk, app usage pattern, and DC of users when they launch apps from their mobile devices.

For the given 900 mobile apps in the permission dataset, we measure the privacy risk of each app, which is one factor of DC, combining the requested permissions and function relativity. The average privacy risks (by (4)) of different mobile app categories are presented in Figure 4. It shows that all the categories are with privacy risk more than 15. According to the statistics shown in Figure 1, one mobile app can access as many as 25 permissions in the dataset, indicating the upper bound of the privacy risk factor $R(a_i)$ is about 60. According to our observation, apps are risky if their privacy risks are over 20 (1/3 of the upper bound). From the figure, the privacy risk expectation of apps for video, audio, web browser, shopping, voice, tools, and security is high. In contrast, apps for news, live streaming, photography, reading, and radio are relatively secure.

To show the impact of function relativity, we conduct the evaluation with varied ω_{ij}, which is weight for function-related and function-unrelated permissions. We fix ω_{ij} to be equal to 1 for function-unrelated permissions and vary ω_{ij} for function-related permissions with 0.2, 0.5, and 0.8 (as the legend) in terms of five-category apps (WiFi, streaming, social, video, and mail). The result as shown in Figure 5 presents the average privacy risk of the five categories of mobile apps. It shows that the trend of privacy risk does not change with different values. Moreover, the higher the value of ω_{ij} chosen for function-related permissions selected, the higher the privacy risk the mobile apps own. Among all the rest of experimental results, we use $\omega_{ij} = 0.5$ for function-related permissions and use $\omega_{ij} = 1$ for the function-unrelated permissions to reflect the different privacy violation concern with respect to function relativity of mobile apps.

The app usage pattern, which is the other factor of DC, measures the time usage of different apps. Figure 6 shows the usage time distribution of all apps with respect to mobile apps usage (Figure 6(a)) and users usage time (Figure 6(b)). They record 900 mobile apps and around 2,000 users who run these 900 apps in the datasets. Both of them follow the heavy tail distribution (see the straight line). Majority of apps have short usage time, while few apps have a long usage time. Similarly, most users use their mobile devices for short time, while few users have long time duration for mobile device usage. To show the usage pattern of specific mobile apps, we select several typical apps (i.e., WeChat, QQ, and Tencent video) and show their usage patterns in Figure 7. The plot shows that over 90% of users run WeChat for less than 10^4 s, while this number of users decreases to 75% for Tencent video. This is due to the different attributes of the apps functionalities, where WeChat is used for messaging, whereas Tencent video is used for video playing. The usage time duration for video playing apps (i.e., Tencent video) should be longer than that for messaging apps (i.e., WeChat and QQ).

We measure the DC of mobile devices according to user-app usage pattern and the risk of mobile apps. Specifically, we evaluate the DC of 50 sampled users given the collected information of 900 mobile apps and draw the DC value of each person. In our experiment, the DC value is bounded by $[0, 60)$. For the convenience of illustration, we ordered the DC values as shown by the line with forward triangles in Figure 8. It shows that all users are with DC value larger than 20, and the highest one reaches almost 30. From our observation, if DC value is over 20, it indicates that the mobile phone usage becomes risky. Our selected 50 users are all in the risky status.

4.3. Evaluation of App Usage Recommendation. To evaluate the performance of the proposed recommendation approach, we calculate DC values of people under different conditions to quantify how much danger of mobile apps with respect to privacy violation to mobile users. Specifically, we vary the parameter b from 0 to 1 and finally to 100, to see the DC turbulence for different users. We compare the proposed

Table 3: The requested permissions of news apps.

Permissions	Baidu	Phoenix	Sohu	Tencent	NetEase	CCTV	Sina	Online retail	The paper
Location		×	×	×	×				
SMS sending				×					
Contacts	×			×					
Network connection	×	×	×	×	×	×	×	×	×
Recording		×	×	×	×				
Camera	×				×				×
System setting	×		×	×	×	×	×		×

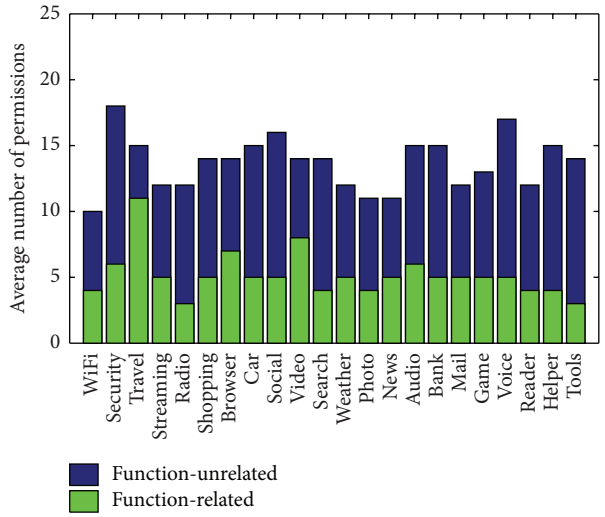


Figure 3: The permission distribution of different categories of apps.

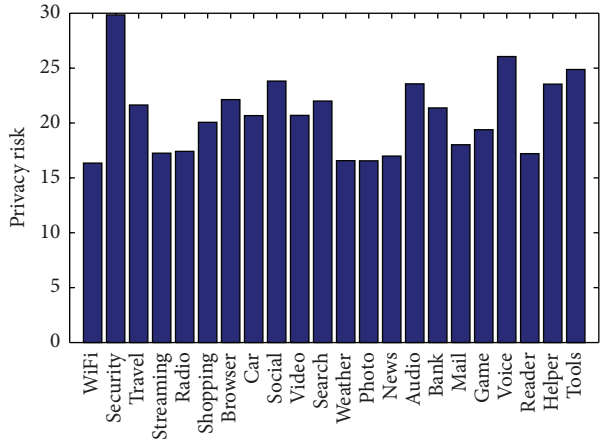


Figure 4: The risk distribution of different categories of apps.

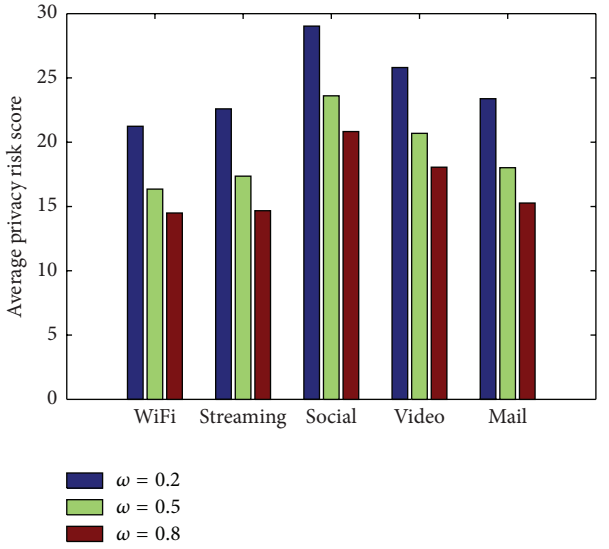


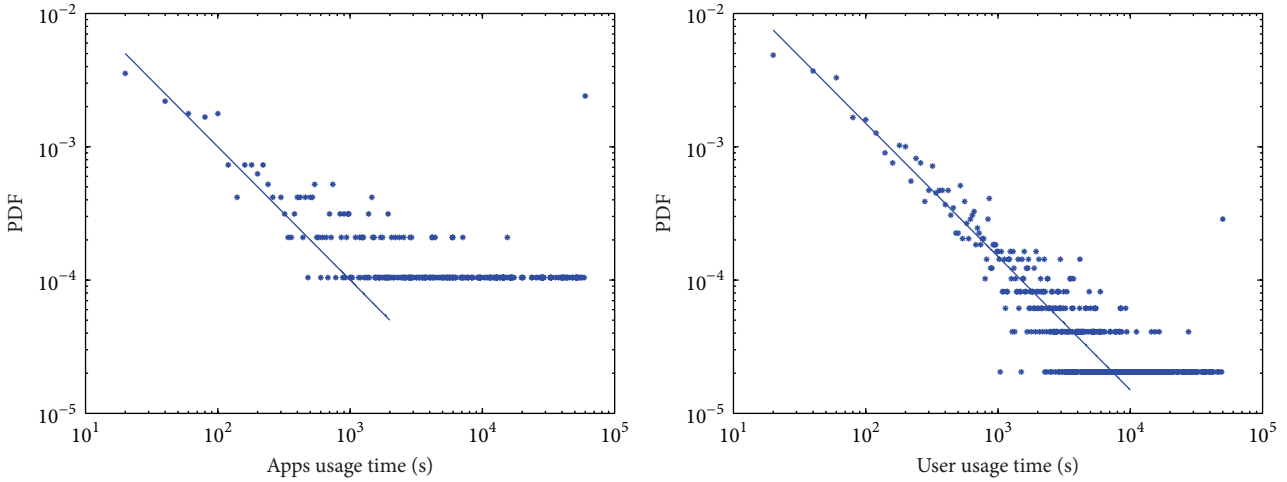
Figure 5: The impact of function relativity as function of ω_{ij}.

approach with the PNB method [10], which only considers the app risk for recommendation. Indeed, PNB provides the baseline for the evaluation.

The result of the evaluation is shown in Figure 8. Besides the ground truth observed in the real world, it presents the DC value of the proposed recommendation approach in case of $b = 0$, $b = 1$, and $b = 100$ and PNB method. When $b = 0$, the recommendation is led by the personalized user-app preference, as illustrated by the line with plus marks. It shows that the danger coefficient is much higher than ground truth if the privacy risk issue is neglected. Meanwhile, it also shows that some users' preference can reach a lower danger coefficient. This is due to the intrinsic low risk of the mobile apps.

Furthermore, we consider the situation in which both user-app preference and privacy have the same importance, which is considered setting $b = 1$, and the line with backward triangles is obtained. The DC value is much lower than the ground truth. Only two users are still with DC value higher than 20. We further increase the value of b to 100 to show the DC values when the risk takes the dominant role. The result is shown as the line with solid circles. It reaches almost the baseline obtained from PNB marked by the line with circles. Without loss of generality, we consider that the majority of people would like to consider preference and privacy equally important. In such case ($b = 1$), the DC values reduce to about 50% on average compared with ground truth.

We further investigate the recommendation results for news app categories. We first show several common permissions of the news apps in Table 3. In all 7 listed permissions, only network connection is the function-related permission



(a) App usage time (b) User usage time

Figure 6: The PDF of usage time in terms of apps and users.

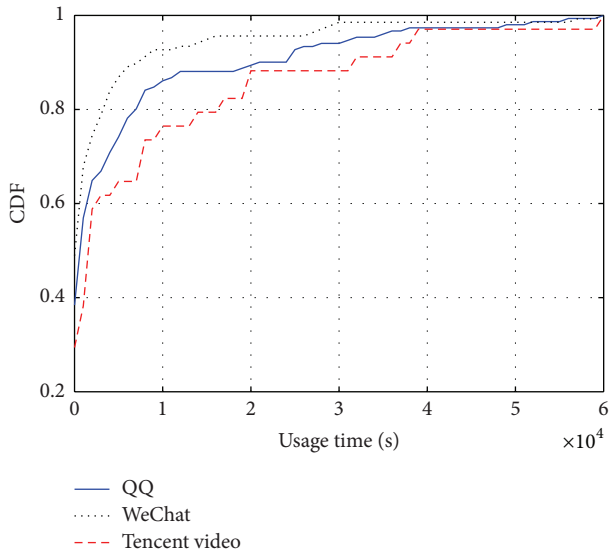


Figure 7: The usage pattern of three mobile apps.

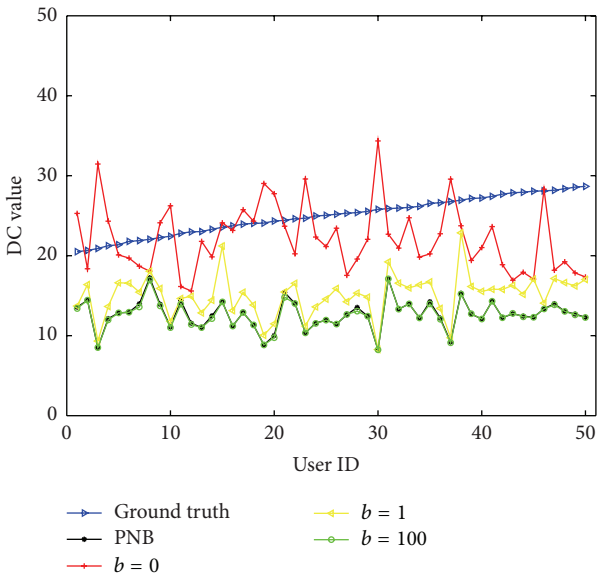


Figure 8: The DC of random selected 50 users.

to news reading. The remaining 6 permissions such as location, SMS, contacts, recording, camera, and system settings are function-unrelated to these apps. For each app, we mark a cross if it asks for a certain permission in the corresponding place. It shows that the app online retail requests only one function-related permission. CCTV and Sina request one function-unrelated permission system setting besides network connection. Baidu, Sohu, NetEase, Phoenix, and the paper request even more function-unrelated permissions. The most severe one is Tencent, which requests nearly all the listed permissions. We show the recommendation list of news apps as a function of different parameters shown in Table 4. When $b = 0$, Baidu News is the first app recommended, indicating it fits user preference for news reading. The paper news is the last one as it is not as popular as the others. When $b = 1$, Baidu News is still in the first place. If people consider privacy and preference equally important, Baidu News is the best choice. However, Tencent News goes to the last position. This is mainly due to its violation of user privacy. If the privacy is considered as the most important factor (when $b = 100$ and PNB), the online retail news app becomes the first recommended app, due to its lowest privacy violation to users, as we have shown in Table 3. In contrast, Baidu News falls into the fourth position. Tencent News still holds the last position, given its high violation to user privacy.

5. Conclusions

In this paper, we proposed a method to evaluate the privacy risk from mobile apps when people use a mobile device. We evaluated the mobile app privacy risk and defined a

Table 4: Recommendation list of news apps.

$b = 0$	Baidu	Phoenix	Sohu	Tencent	NetEase	CCTV	Sina	Online retail	The paper
$b = 1$	Baidu	CCTV	Online retail	Sina	Phoenix	Sohu	NetEase	The paper	Tencent
$b = 100$	Online retail	CCTV	Sina	Baidu	NetEase	Sohu	Phoenix	The paper	Tencent
PNB	Online retail	CCTV	Sina	Baidu	NetEase	Sohu	Phoenix	The paper	Tencent

danger coefficient for each user by combining the mobile apps risk and user preference. According to the requirement of privacy and satisfaction of app usage preference, we proposed a mobile app recommendation method named AppURank. The evaluation results showed that the privacy risks of apps are different, and the DC of mobile users is very high for many of them. The proposed recommendation method can help to reduce the danger coefficient by 50% on average and meanwhile maintains personalized user preference. For the future work, we will build the mobile app recommendation system on the mobile device and evaluate the performance of the recommendation algorithm by implementation and deployment.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work has been partially sponsored by National Science Foundation of China (no. 61502045), EU FP7 IRSES Mobile Cloud Project (Grant no. 612212), the 111 Project (no. B08004), the Fundamental Research Funds for the Central Universities, and the Beijing Higher Education Young Elite Teacher Project.

References

[1] B. Yan and G. Chen, "AppJoy: personalized mobile application discovery," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys '11)*, pp. 113–126, ACM, Washington, DC, USA, July 2011.

[2] K. Yu, B. Zhang, H. Zhu, H. Cao, and J. Tian, "Towards personalized context-aware recommendation by mining context logs through topic models," in *Advances in Knowledge Discovery and Data Mining: 16th Pacific-Asia Conference, PAKDD 2012, Kuala Lumpur, Malaysia, May 29-June 1, 2012, Proceedings, Part I*, vol. 7301 of *Lecture Notes in Computer Science*, pp. 431–443, Springer, Berlin, Germany, 2012.

[3] H. Zhu, E. Chen, H. Xiong, K. Yu, H. Cao, and J. Tian, "Mining mobile user preferences for personalized context-aware recommendation," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 4, article 58, 2015.

[4] H. Zhu, H. Xiong, Y. Ge, and E. Chen, "Mobile app recommendations with security and privacy awareness," in *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '14)*, pp. 951–960, ACM, New York, NY, USA, August 2014.

[5] 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 (CCS '11)*, pp. 627–638, ACM, Chicago, Ill, USA, October 2011.

[6] K. W. Y. Au, Y. F. Zhou, Z. Huang, P. Gill, and D. Lie, "Short paper: a look at smartphone permission models," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices (SPSM '11)*, pp. 63–67, ACM, October 2011.

[7] W. Enck, P. Gilbert, B.-G. Chun et al., "An information-flow tracking system for realtime privacy monitoring on smartphones," in *Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation (OSDI '10)*, pp. 1–6, USENIX Association, Berkeley, Calif, USA, 2010.

[8] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, "Taming information-stealing smartphone applications (on android)," in *Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST '11)*, pp. 93–107, Springer, Pittsburgh, Pa, USA, June 2011.

[9] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS '09)*, pp. 235–245, ACM, Chicago, Ill, USA, November 2009.

[10] H. Peng, C. Gates, B. Sarma et al., "Using probabilistic generative models for ranking risks of Android apps," in *Proceedings of the ACM Conference on Computer and Communications Security (CCS '12)*, pp. 241–252, October 2012.

[11] J. Lin, B. Liu, N. Sadeh, and J. I. Hong, "Modeling users' mobile app privacy preferences: restoring usability in a sea of permission settings," in *Proceedings of the Symposium on Usable Privacy and Security (SOUPS '14)*, pp. 199–212, USENIX Association, Menlo Park, Calif, USA, July 2014.

[12] R. Liu, J. Cao, L. Yang, and K. Zhang, "PriWe: recommendation for privacy settings of mobile apps based on crowdsourced users' expectations," in *Proceedings of the IEEE International Conference on Mobile Services (MS '15)*, pp. 150–157, New York City, NY, USA, June 2015.

[13] S. K. Katti and A. V. Rao, "Handbook of the poisson distribution," *Technometrics*, vol. 10, no. 2, pp. 412–412, 1968.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *The Journal of Machine Learning Research*, vol. 3, no. 4-5, pp. 993–1022, 2003.