PrivMetrics: A Framework for Quantifying User Privacy in Smartphones

Suranga Seneviratne[†⋆], Aruna Seneviratne[†⋆], Johan Kestenare[†]

[†]NICTA, Australia; [⋆]School of EET, University of New South Wales, Australia

email : firstname.lastname@nicta.com.au

ABSTRACT

Smartphones are becoming ubiquitous and the associated ecosystem is dominated by third party apps as they are required to get the full benefits of a smartphone. A popular method of monetizing these third party apps, specially the free apps, is through advertising and providing users' personal data to analytic services (trackers). The collection of users' private information for advertisements and to analytics in general is problematic because, it might lead to privacy violations. However, at present there are no mechanisms for users to determine the implications of third-party apps collecting their personal information. In this paper, we propose *PrivMetrics* framework which enables users to make informed decisions about the use of these third party apps by providing them with an analysis on their privacy leakages and wherever possible recommending alternative apps with the same functionality as the apps they are considering which better protect their overall privacy.

1. INTRODUCTION

Smartphone usage is driven by the availability of third party apps. In parallel to the growth of these third party apps, especially the free apps, a large diverse ecosystem of organizations that collect and aggregate information about the users of these apps have emerged. These organizations (*trackers*), use information about users for various purposes, such as providing personalized services, or sending them targeted advertisements. Numerous research groups have exposed the privacy leakages that occur as a result of developers using users' personal data to monetize the free apps by integrating *tracker* libraries with the app [5, 8]. This has lead to the development of tools and frameworks to assess the privacy leakages of individual apps [3, 12].

In the current smartphone app ecosystem, there are only a limited number of *trackers*. Therefore the user information collected from various individual apps, will be collected by only a few *trackers*. This allows these few *trackers* to build accurate profiles of the users and infer various attributes such as age, gender, marital status, and health conditions which the users may not want to be known [2, 10, 14, 11]. Thus to assess privacy implications for a user, it is necessary analyze the overall flow of information to the *trackers* in addition to analyzing individual apps. Currently, there is no easy way of knowing what information an individual app is collecting. Moreover, even if a user did get to know that a given app is collecting personal information, there is no way of knowing the overall privacy implications, and there are only limited options other than not using the app to avoid the loss the privacy. In addition, as some of the information may have been collected at the time of installation, just not using the app or uninstalling app might also be too late.

This paper presents *PrivMetrics*, a framework that addresses the above challenges by analyzing individual apps offline and providing users with recommendations about the overall privacy implications of installing a new app. In addition, it provides users with recommendations of other apps that have similar functionality, but is less privacy intrusive, through a novel algorithm based on *steepest ascent hill climbing*. We present the preliminary results of app recommendations using a dataset collected from over 300 smartphone users and show that 50% of the users, were able to achieve more than 20% of improvement in their *overall privacy level* by using *PrivMetrics*.

The remainder of this paper is organized as follows. In Section 2 we present the design goals of *PrivMetrics* and Section 3 describes the *PrivMetrics* architecture. Algorithm proposed for improving the *overall privacy level* is presented in Section 4. The preliminary results are presented in Section 5 and Section 6 concludes the paper.

2. DESIGN GOALS

The overarching goal of *PrivMetrics* is to inform the users of the potential privacy implications of installing a third party app and where possible indicate possible alternative apps with similar functionality, that can have a lower impact on their privacy. This is realized by designing the system with the following sub goals:

- **Provide details to the users on what data is being collected and the privacy implications**

It is a common practice to list the permissions required by the app when a user tries to install an app . However, these permissions are abstract and most users do not understand them. In addition, some of the data that is collected, for example collecting the list of apps already installed on the smartphone, does not require any permission from the user. Thus, it is necessary to provide users with an easily understandable explanation about the implication of the information that is being requested.

- **Provide user the control of their personal data**

  The objective of *trackers* is to infer user traits from the data that is collected which will enable the provision of personalized services and other activities such as displaying tailored advertisements. From the users' point of view, while sharing some traits might be acceptable, others might not be. For example, one user will be willing to share the location information with third parties, however might not be willing to share gender information. Thus, it is necessary to provide the user control of data that will give them the option of determine which information they are not willing to share and evaluate the suitability of apps based on their preferences.

- **Provide users with alternative app recommendations**

  As there are only a limited number of trackers, data collected from different apps can be sent to the same *tracker*. As a result, when a new app is installed by the user, it may lead to a significant loss of privacy. It is important to assess what information is being collected and the impact of this information on their privacy by taking into account the information that is being collected by the already installed apps and which trackers they are being sent to. For example, an app already installed may provide a given *tracker* the user's location information. Thus, if the user installs an app which only asks for the gender (either directly or by accessing a social network profile) and provides the information to the same tracker, individually neither apps violate the requirement of location and gender should not be provided together. However, as soon as the second app is installed, it violates the requirement. Therefore, it is necessary to warn the user as well as provide, where possible, an alternative application that has the same functionality, before the user installs the app.

3. PrivMetrics ARCHITECTURE

Figure 1 presents the overall architecture of *PrivMetrics*. It consists of three phases, which are described below.

1. **App discovery:** In this phase, *PrivMetrics* discovers the available apps by crawling the app markets. Then, it downloads the executables of the discovered apps, and stores them in an database for offline analysis.

2. **App classification:** In this phase, *PrivMetrics* analyzes each of the downloaded apps *statically* and *dynamically*, and for each app determines

   - ***The private data that is being collected and trackers those data being shared with*** There are various open and commercial tools to do this both statically and dynamically. Static analysis tools can decompile the applications and identify the requested permissions [1, 7]. Once the code is disassembled it is possible to identify whether the private data access is done by the first party or the third party by observing the library names. Dynamic analysis tools [1, 9, 13, 6] execute the apps in controlled environments and provide further insights to data leakage by highlighting the websites or domains data is being sent.

   - ***The permissions requested against the app functionality*** Some apps might collect information without tracking libraries through over-permission *(i.e. asking for permissions which are not required for the core app functionality)*. Those apps are identified using machine learning approaches [4]. First apps are clustered according to the functionality by mining the app descriptions and reviews. Then most commonly requested permissions are identified for a given cluster. If an application belonging to a given cluster requests permissions that are uncommon that cluster, they are flagged as requesting *over-permissions*.

3. **App recommendation** Once the app classification is done, *PrivMetrics* during this phase calculates the existing overall level of privacy for a given user. The user's requirements with respect to sharing personal information is taken into account as described in the Section 4. Then it recommends the apps that provide similar functionality, but improves the overall privacy.

An example, operation of *PrivMetrics* is schematically shown in Figure 2. In this scenario, the user has

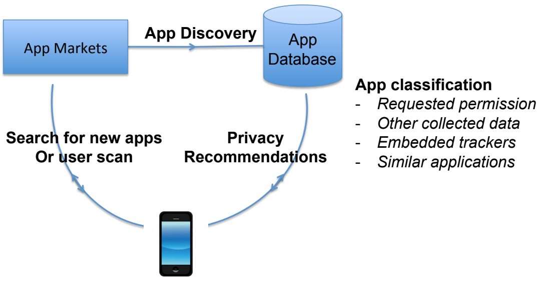

Figure 1: Basic architecture of *PrivMetrics*

3 apps: *Spam SMS filter, Transit schedule* and an *Arcade game* and all of the three apps are connected to the same *tracker*, Tracker1. This gives Tracker1 access user's *SMS content* and *Location*. By analysing similar apps, *PrivMetrics*recommends the user to consider using an alternative Spam SMS filter app which does not share the SMS content with any *tracker*.

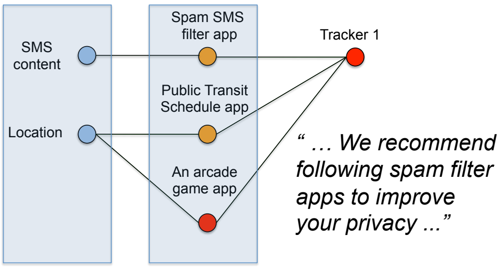
SMS content
Location
Spam SMS filter app
Public Transit Schedule app
An arcade game app
Tracker 1
" ... We recommend following spam filter apps to improve your privacy ..."

Figure 2: Example user case of *PrivMetrics*

As there are tools and techniques for ***App discovery*** and ***App classification*** as described previously, the rest of the paper focuses on ***App recommendation***, which allows personalized control on the data leakage by suggesting alternative apps.

4. PrivMetrics RECOMMENDATION

The parameters and the notation used in main algorithm in *PrivMetrics* for app recommendation is shown in Figure 3.

It shows apps on a user device $A_1, A_2, ..., A_K$ and each of the K apps, is connected one or more of the N data sources, S_n and D trackers, Y_d, shown as edges in the graph. We assume that each tracker is trying to predict P user traits, U_p, using the information obtained through data sources. The accuracy a tracker, Y_d can predict the user trait U_p using the available data sources is denoted by $Pr_{Y_d}(U_p)$.

We assume the relationship between access to data sources (S_i) and accuracy predicting user trait, U_p is given by the function φ. For simplicity we also assume that the predicting capability of all the *trackers* are the same. However the model is flexible to support different prediction capabilities. Then

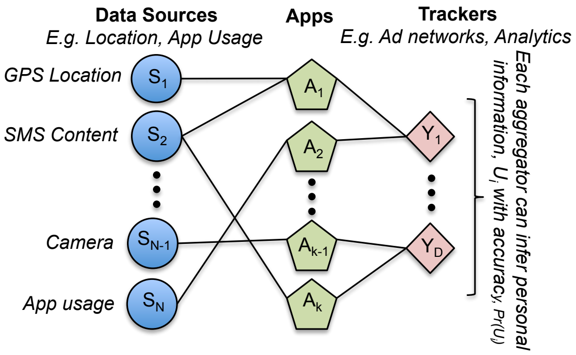

Figure 3: Graph representing the interconnection between data sources and trackers for a given user

$$\varphi_{U_p}(S_1, S_2, ..., S_N) = Pr(U_p) \; \forall i = 1 : P$$

The function φ is at this stage is unknown and requires further study. To demonstrate the potential of the proposed framework, initially we assume φ be a *linear function*. i.e.,

$$\varphi_{U_p} = \alpha_0 + \alpha_1 s_1 + ... + \alpha_M s_M$$

where s_m is a binary variable representing the access to the data source S_m. Thus we have a predictablity vector Φ_{Y_d} for each of the tracker of size P which is given by:

$$\Phi_{Y_d} = (Pr(U_1) \; Pr(U_2) \; ... \; Pr(U_P))^T$$

We then define a user willingness vector, populated by the user, which indicates whether the user is willing enable a tracker to infer a trait given by:
$\Lambda = (\lambda_1 \; \lambda_2 \; ... \; \lambda_P)^T$ of size P,
Then we calculate the *Privacy level* with tracker Y_d,

$$f_{Y_d} = \sum_{L=1}^{l} (1 - Pr(U_l))/l$$

where U_l are the traits which user marked as concerned in Λ.

The *overall privacy level*, G for the user considering all trackers, is defined as the mean of the *Privacy levels* with each tracker, given by:

$$G = \sum_{d=1}^{D} f_{Y_d}/D$$

For some of the applications there can be multiple other applications in the market which have the same functionality. If there are, it may be possible to use one these alternative applications rather than one being considered or installed to improve the *overall privacy level* of a user. This can be achieved by solving the below optimization problem.

$$\underset{A_1,...,A_J}{\text{maximize}} \quad G(A_1, ..., A_J) \text{ subject to } A_j \in \{A_{\{}j, c_j\}\}$$

Here A_{j,c_j} is the set of similar applications to A_j of size c_j.

This is done by using an variant of the *steepest ascent hill climbing* algorithm. The details of this algorithm used to generate the alternative apps recommendations is given below.

```
Data:
Initial set ← $\{A_1, A_2, ..., A_K\}$
Similar app sets ← $\{\{A_{1,c_1}\}, \{A_{2,c_2}\}, ..., \{A_{K,c_K}\}\}$
Result:
Optimized $G_o$
Optimized set ← $\{A_1, A_2, ..., A_J\}$
Initialization;
Calculate initial $G_i$;
for $k = 1$ to $k = K$ do
    Optimized set←Initial set;
    $G_o$←$G_i$;
    for $j = 1$ to $j = c_j$ do
        Current set ← $\{A_1, A_2, ..., A_K\}$ where $A_k$
        replaced by $A_j$;
        Current G, $G_c$ ←G(Current Set);
        if $G_c \geq G_o$ then
            $G_i \leftarrow G_c$;
            Initial set←Current set;
        end
    end
end
Return
$G_o$
Optimized set;
```

Algorithm 1: Steepest ascent hill climbing algorithm for optimizing global privacy level of a user

5. PRELIMINARY RESULTS

We tested the *PrivMetrics* framework on a dataset collected from smartphone users which is used in our previous work [10]. The dataset contained lists of apps installed by 339 Android smartphone users and for each app we crawled the Google Play Store and identified similar apps using the similar apps feature available in the store. For each app we selected the first 3 similar apps recommended by Google as the possible replacements. We downloaded the *apk* files from Google Play Store for these apps and the remaining analysis is based on 2072 apps we have downloaded at the time of submission. For each app, we identified data sources it accesses and the embedded tracking companies by decompiling the *apk* files.

We applied our algorithm to optimize the *global privacy level* of each user. Figure 4 shows the CDF of the percentage improvement we were able to achieve. As can be seen for 50% of the users, we were able to achieve more than a 20% increase and a further 20% of the users we were able to achieve more than a 50% increase. Figure 5 shows an example scenario. For this particular scenario, *PrivMetrics* proposed the user to replace *AirDroid* with *Android device manager* (both are android device manager apps) and *Twitter* with *Twittercaster* (an unofficial *Twitter* client) that led to an improvement of 10% in the overall privacy level.

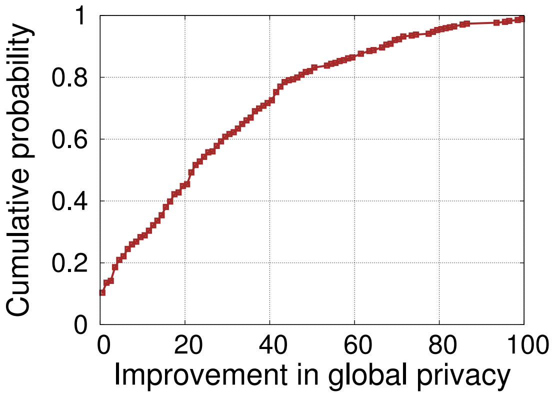

Figure 4: CDF of % improvment

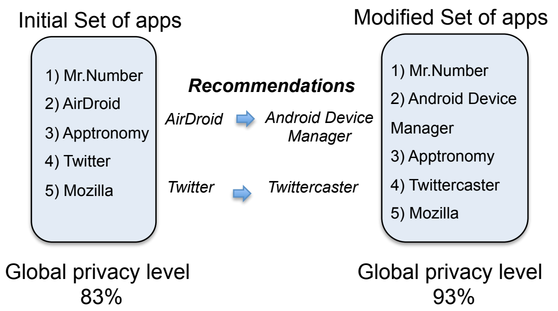

Figure 5: Example user

6. CONCLUSION AND FUTURE WORK

With the increase in the availability of third party apps, and the domination of the market by a few *tracking* companies, there is a steady erosion of user privacy. Therefore, it is necessary to develop methods that will provide user more control. *PrivMetrics* provides such a framework. The potential of *PrivMetrics* was demonstrated, using a real dataset and some simpliflied assumptions. Despite these simplified assumptions, the results are very encouraging and shows the viability of the *PrivMetrics* framework. We intend to improve the framework by addressing the following issues.

- The model used for determining the prediction capability of trackers by analysing the dependancies of other information such as frequency of data collection and the information already present with aggregators.
- The *similar app identification* method by analysing the suitability of the recommendations and identifying the means of improving the current algorithm such as ranking apps so that the algorithm converges to a improved local maximum.

7. REFERENCES

[1] S. Arzt, S. Rasthofer, C. Fritz, E. Bodden, A. Bartel, J. Klein, Y. Le Traon, D. Octeau, and P. McDaniel. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. In *Proc. of the 35th ACM SIGPLAN*, page 29. ACM, 2014.

[2] G. Chittaranjan, B. Jan, and D. Gatica-Perez. Who's who with big-five: Analyzing and classifying personality traits with smartphones. In *Proc. - ISWC*, pages 29–36, 2011.

[3] W. Enck, P. Gilbert, B. G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. Taintdroid: an information flow tracking system for real-time privacy monitoring on smartphones. *Communications of the ACM*, 57(3):99–106, 2014.

[4] A. Gorla, I. Tavecchia, F. Gross, and A. Zeller. Checking app behavior against app descriptions. In *ICSE*, pages 1025–1035, 2014.

[5] M. Grace, W. Zhou, X. Jiang, and A. Sadeghi. Unsafe exposure analysis of mobile in-app advertisements. In *Proc. of the 5th ACM WiSec*, pages 101–112, 2012.

[6] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 639–652. ACM, 2011.

[7] AVG Threat Labs. Inc. Website safety ratings and reputation. `http://www.avgthreatlabs.com/website-safety-reports/app`, 2014.

[8] I. Leontiadis, C. Efstratiou, M. Picone, and C. Mascolo. Don't kill my ads!: balancing privacy in an ad-supported mobile application market. In *Proceedings of the Twelfth Workshop on Mobile Computing Systems & Applications*, page 2. ACM, 2012.

[9] Joe Security LCC. Joe Sandbox Mobile. http://www.joesecurity.org/joe-sandbox-mobile, 2014.

[10] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti. Predicting user traits from a snapshot of apps installed on a smartphone. *ACM SIGMOBILE Mobile Computing and Communications Review*, 18(2):1–8, 2014.

[11] S. Seneviratne, A. Seneviratne, P. Mohapatra, and A. Mahanti. Your installed apps reveal your gender and more! In *Proceedings of the Workshop on Security and Privacy in Mobile Environments (SPME 2014)*. ACM, 2014.

[12] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos. Profiledroid: multi-layer profiling of android applications. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 137–148. ACM, 2012.

[13] L. Weichselbaum, M. Neugschwandtner, M. Lindorfer, Y. Fratantonio, V. van der Veen, and C. Platzer. Andrubis: Android malware under the magnifying glass. *Vienna University of Technology, Tech. Rep. TRISECLAB-0414-001*, 2014.

[14] J. J. C. Ying, Y. J. Chang, C. M. Huang, and V. S. Tseng. Demographic prediction based on users mobile behaviors. *Mobile Data Challenge*, 2012.