SOFTWARE—PRACTICE AND EXPERIENCE
Softw. Pract. Exper. 2015; **00**:1–26
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/spe

Privacy-preserving targeted mobile advertising: Requirements, design, and a prototype implementation

Yang Liu[1]*and Andrew Simpson[1]

[1]*Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK*

SUMMARY

With the continued proliferation of mobile devices, the collection of information associated with such devices and their users — such as location, installed applications and cookies associated with built-in browsers — has become increasingly straightforward. By analysing such information, organisations are often able to deliver more relevant and better focused advertisements. Of course, such targeted mobile advertising gives rise to a number of concerns, with privacy-related concerns being prominent. In this paper, we discuss the necessary balance that needs to be struck between privacy and utility in this emerging area and propose *PPTMA* (Privacy-Preserving Targeted Mobile Advertising) as a solution that tries to achieve that balance. Our aim is to develop a solution that can be deployed by users, but is also palatable to businesses that operate in this space. This paper focuses on the requirements and design of *PPTMA*, and also describes an initial prototype. We also discuss how more detailed technical aspects and a complete evaluation will underpin our future work in this area.
Copyright © 2015 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: targeted advertising; privacy; mobile devices; Android

1. INTRODUCTION

Online Targeted Advertising (OTA), also known as *Online Behavioural Advertising* (OBA), has brought significant benefits to advertisers since it was first proposed. Although OTA leverages some relatively new technologies, the concept of ‘behavioural advertising’ is not new: marketers have collected customers’ data to tailor advertising messages to specific audiences since at least the 1960s [1]. In the context of online commerce, DoubleClick began to track and analyse users’ browsing patterns using third-party cookies for the purposes of presenting targeted advertisements in the late 1990s [2]. Using the information collected from users’ online behaviour, including historical search queries, site views and other operations on particular types of content, advertisers are able to deliver the most relevant advertisements (ads) to their users. These ads can match users’ interests with a high level of accuracy; as such, OTA improves the advertising effectiveness significantly.

It is clear that TMA is becoming a vital marketing method for a wide range of companies. To apply TMA, companies need to track and analyse users’ operations to create interest profiles for each of them. With these profiles, suppliers of ads could display related ads to those people who are potentially interested in particular goods. In order to obtain more detailed consumers’ personal information, some advertising companies choose to form an advertising alliance [3] or to cooperate

*Correspondence to: Yang Liu, Department of Computer Science, University of Oxford, Wolfson Building, Parks Road, Oxford OX1 3QD, UK.
E-mail: Yang.Liu@cs.ox.ac.uk

with service platforms run by, for example, Google and Facebook. For example, Google AdSense [4] allows publishers in the Google network of content sites and apps to serve targeted ads.

Users' personal information plays a key role in targeting; as such, it is only to be expected that users will have concerns about the use of their personal information. Although TMA promises an improvement over traditional delivery approaches, the significant privacy concerns associated with it have given rise to a degree of hostility towards advertisers. A number of studies (see, for example, the contributions of [5] and [6]) indicate a direct relationship between consumer attitudes and behaviour: consumers' trust in privacy of mobile advertising is positively related to their willingness to accept mobile advertising, and they generally have negative attitudes towards mobile advertising unless they have specifically consented to it.

The relationship between consumers and advertisers is a delicate one and needs to be handled appropriately: a solution to the privacy-preserving targeted mobile advertising 'problem' should provide benefits both to consumers — so that they can take advantage of ads without worrying about their privacy being compromised — and to the companies — resulting in reduced hostility and increased response rates. Previous work in this area has tended to take one 'side' or the other: those on the 'side' of consumers might propose solutions that deceive advertisers; those on the 'side' of companies might propose solutions that bypass privacy concerns. We aim to develop a solution that strikes a balance between concerns and has the potential to be palatable to both 'sides'. Thus, this paper is concerned with the requirements for, and the design of, what we term *Privacy-Preserving Targeted Mobile Advertising* (*PPTMA*).

Section 2 provides the motivation for our contribution. Then, in Section 3, we present our requirements. In Section 4 we give consideration to some threats to privacy. The architecture and design of our solution is presented in Section 5. We discuss our prototype solution in Section 6. In Section 7 we place our work into context by discussing related work. Finally, in Section 8, we summarise the contribution of this paper and give consideration to possible areas of future work.

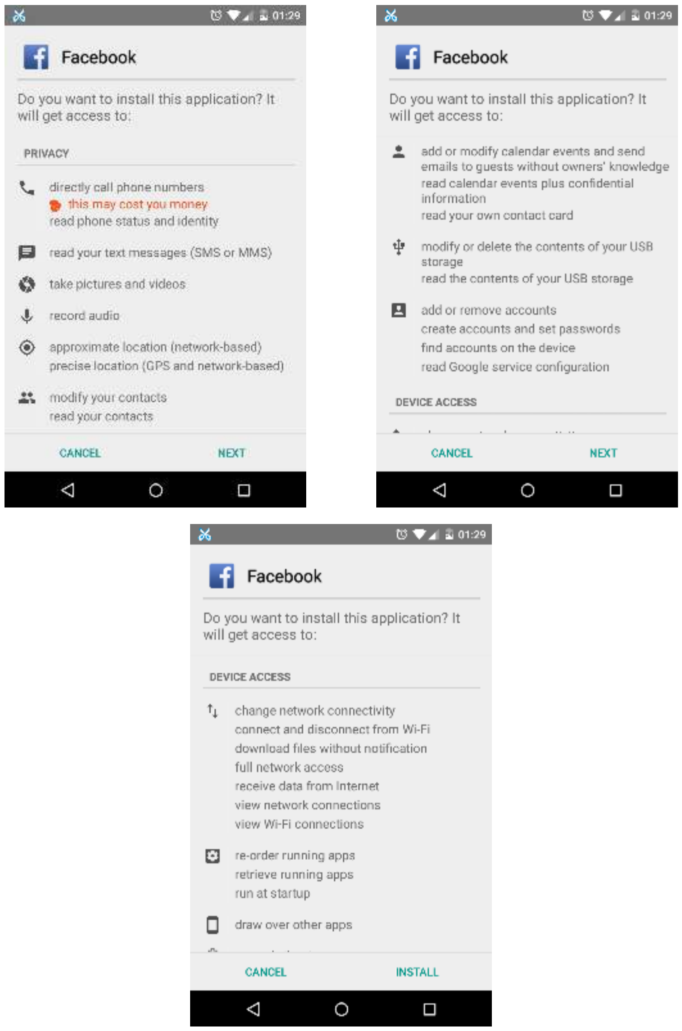
2. MOTIVATION

The privacy issues related to TMA are inherently complex: there are many parties and roles involved; there is a need to understand conflicting user attitudes and behaviour; and there are various types of personal data to be protected. In this section, we provide a brief overview of the key issues with a view to providing context for our contribution. We begin by giving consideration to what some authors have described as the 'privacy paradox'.

2.1. A Privacy Paradox: online users' attitudes and behaviour

Many authors, including Awad and Krishnan [7], Barnes [8], Norbert *et al.* [9], and Utz and Krämer [10], have given consideration to the term *privacy paradox*, which recognises the inconsistency between people's reported attitudes and their observed behaviour when interacting with online services. As social exchange theory [11] explains, if the exchange is perceived to be beneficial, then the individual is likely to enter into an exchange relationship. With regard to TMA, users exchange the benefits of useful mobile online ads with their privacy. Sometimes consumers' unwillingness to pay for privacy can be rationalised with relatively small rewards. For example, a field experiment [12] involving two stores showed that participants are willing to provide personal information such as their monthly income and date of birth for only a 1 euro discount. Surprisingly, and inconsistent with their behaviour, about 82% of the participants reported a decreased willingness to provide their income information. In fact, participants predominantly chose to shop in the store with the lower price but the requirement for more sensitive data. Moreover, the experiment also shows that even though prices, shopping time and delivery time were equal at both stores, the store that captured less personal information failed to attract more customers.

The great protections claimed by privacy notices also contribute to users' relatively lax privacy attitudes, in which case privacy violations can be caused by users' over reliance on privacy notices. Martin [13] suggests that users have been found to assume the protections offered by privacy notices


01:29
Facebook
Do you want to install this application? It will get access to:
PRIVACY
directly call phone numbers
this may cost you money
read phone status and identity
read your text messages (SMS or MMS)
take pictures and videos
record audio
approximate location (network-based)
precise location (GPS and network-based)
modify your contacts
read your contacts
CANCEL NEXT

01:29
Facebook
Do you want to install this application? It will get access to:
add or modify calendar events and send emails to guests without owners' knowledge
read calendar events plus confidential information
read your own contact card
modify or delete the contents of your USB storage
read the contents of your USB storage
add or remove accounts
create accounts and set passwords
find accounts on the device
read Google service configuration
DEVICE ACCESS
CANCEL NEXT

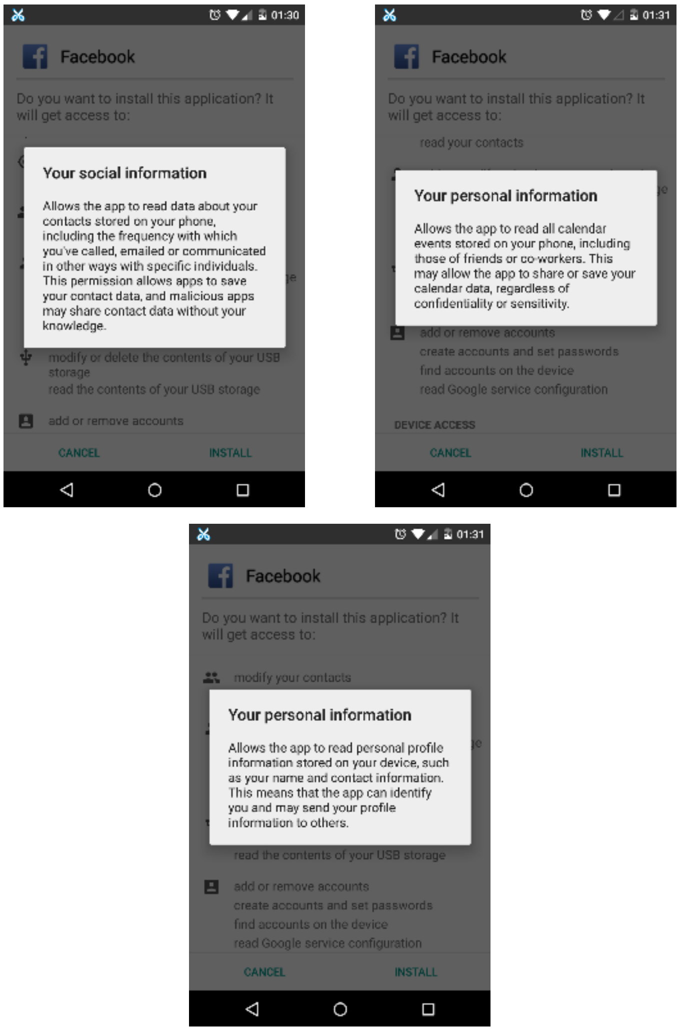
01:29
Facebook
Do you want to install this application? It will get access to:
DEVICE ACCESS
change network connectivity
connect and disconnect from Wi-Fi
download files without notification
full network access
receive data from Internet
view network connections
view Wi-Fi connections
re-order running apps
retrieve running apps
run at startup
draw over other apps
CANCEL INSTALL

Figure 1. Facebook permissions list

can meet their privacy expectations even when some stated notices differ considerably from the actual ones. Further, privacy notices are likely to be ignored due to the long text and obscure language [14, 15]. As a result, consumers may adopt a blind reliance on firms' business ethics.

On the other hand, research undertaken by Tucker [16, 17] suggests that the opposite is true: that consumers still pay attention to their personal information while using online services. An experiment undertaken by Tucker [17] showed that a Facebook page was twice as effective at attracting users after the Facebook policy gave users more control over their own personal information and made the privacy settings easier to use. The experiment could imply that, by giving consumers explicit control over how their data will be used, the advertisers may be able to alleviate some of the hostile awareness caused by violating users' privacy while performing OTA.

The ways in which online advertisers collect users' personal information can be classified into two categories according to different initiators: the first is when advertisers capture and analyse users' behaviour or collect context variables from various sensors to obtain personal information; the second is when users submit personal information voluntarily. Most data collected in the first way can be presumed to be real and objective; information obtained in the second way is less likely to be authentic, as users could offer fake data. Several authors have proposed surveys to determine how truthful information returned is. For example, in 2005, Marwick [18] randomly selected 30 user profiles from three different social networking sites (Friendster, MySpace and Orkut) and analysed


Facebook

Do you want to install this application? It will get access to:

Your social information

Allows the app to read data about your contacts stored on your phone, including the frequency with which you've called, emailed or communicated in other ways with specific individuals. This permission allows apps to save your contact data, and malicious apps may share contact data without your knowledge.

modify or delete the contents of your USB storage
read the contents of your USB storage

add or remove accounts

CANCEL INSTALL

Facebook

Do you want to install this application? It will get access to:

read your contacts

Your personal information

Allows the app to read all calendar events stored on your phone, including those of friends or co-workers. This may allow the app to share or save your calendar data, regardless of confidentiality or sensitivity.

add or remove accounts
create accounts and set passwords
find accounts on the device
read Google service configuration

DEVICE ACCESS

CANCEL INSTALL

Facebook

Do you want to install this application? It will get access to:

modify your contacts

Your personal information

Allows the app to read personal profile information stored on your device, such as your name and contact information. This means that the app can identify you and may send your profile information to others.

read the contents of your USB storage

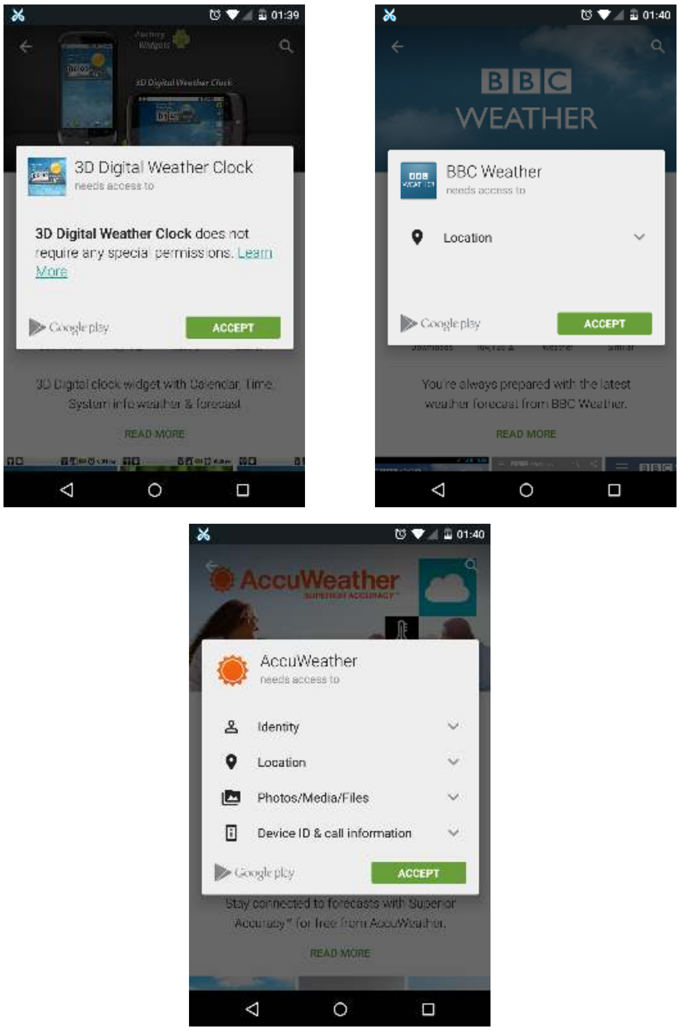
add or remove accounts
create accounts and set passwords
find accounts on the device
read Google service configuration

CANCEL INSTALL

Figure 2. Permission details

the pictures, text and testimonials on each. The result showed that the great majority of user profiles (72 out of 90 — 80%) were presumed to be authentic. Although the sample was relatively small, the study did demonstrate some characteristics of the users of online services. In 2008, Tufekci [19] used a much larger sample (n = 704) of college students to discuss the information disclosure in online social networks. A significant majority (94.9%) of the sample said that they used their real name on Facebook, although there is no obligation to submit their real information. A report [20] in 2012 also demonstrates the phenomenon. Researchers asked 802 teenaged social media users about ten different categories of personal information that they might post on the profile they use most often and found that: 92% of them post their real name; 91% of them post a photo of themselves; 84% of them post their real interests, such as movies, music or books they like; 82% of them post their birth date; and 71% of them post the city where they live. These studies indicate that the great majority of online users submit real personal information to the service suppliers voluntarily, which provides detailed data for advertisers to apply TMA. The lack of awareness of privacy risks might account for this phenomenon. For instance, in [20] it is reported that teenaged users do not express a high level of concern about third party access to their data: only 9% say they are 'very concerned'.

The privacy paradox also exists in the TMA context. To enjoy useful services — using free applications, obtaining various coupons, tracking interested products, etc. — many users choose to compromise: they provide personal information to advertisers (actively or passively), while


3D Digital Weather Clock
needs access to

3D Digital Weather Clock does not require any special permissions. Learn More

Google play ACCEPT

3D Digital clock widget with Calendar, Time, System info weather & forecast

READ MORE

BBC Weather
needs access to

Location

Google play ACCEPT

You're always prepared with the latest weather forecast from BBC Weather.

READ MORE

AccuWeather
needs access to

Identity

Location

Photos/Media/Files

Device ID & call information

Google play ACCEPT

Stay connected to forecasts with Superior Accuracy™ for free from AccuWeather.

READ MORE

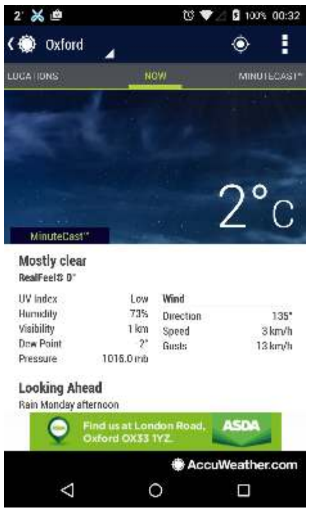
Figure 3. Permission lists of three weather apps

maintaining that privacy is important to them. At the same time, advertisers have recognised consumers' increasing hostility, but TMA is so vital that they cannot easily give up collecting personal information. We suggest that a new approach might help to reconcile the relationship between consumers and advertisers. To this end, we start by considering a motivating example.

2.2. *A motivating example*

Android apps must declare a list of exact permissions they require to execute — for example, whether the app needs to read the user's contacts, obtain the user's location information, or modify the user's calendar; this list shows what kinds of access the application has to the user's personal information. These permissions must be declared in the manifest file of the app, and are presented to the user in the first stage of the installation procedure. For example, upon the installation of Facebook (version 27.0.0.25.15), the user is presented with a list showing that the application requires 39 permissions, 18 of which are related to privacy and security (see Figure 1). When clicking on an item of that list, a more detailed description is given (see Figure 2). Having been presented with this information, the user can then decide whether to install the application.

The approach enables users to determine whether an app is likely to demonstrate unwanted behaviour. Three weather apps are shown in Figure 3. The 3D Digital Weather Clock does not



Mostly clear
RealFeel® 0°

UV Index	Low	Wind	
Humidity	73%	Direction	135°
Visibility	1 km	Speed	3 km/h
Dew Point	2°	Gusts	13 km/h
Pressure	1015.0 mb		

Looking Ahead
Rain Monday afternoon

Find us at London Road, Oxford OX33 1YZ. ASDA

AccuWeather.com

Figure 4. An app with targeted advertisements

require any special permissions; the BBC Weather app asks for the user's location, which is used to provide localised forecasts. The AccuWeather app, however, requires the user's identity, the Device ID and call information. None of this is relevant for weather prediction, but is useful for, for example, targeted advertising. Based on this information, users can make informed decisions as to which of the apps they might wish to install. It is worth noting, however, that the reality does not match our expectations. In fact, at the end of January 2015, there were approximately 65,000 5-star comments for the 3D Digital Weather Clock, but more than 599,000 5-star comments for AccuWeather, which requires the most permissions of the three.[†] AccuWeather does feature targeted mobile ads (see Figure 4, where an advertisement for a local Asda supermarket is presented to an Oxford-based user), and its popularity may very well be due to its functionality: it provides a user-friendly interface and powerful extra functions. Nevertheless, many users have submitted negative ratings on the basis of privacy and targeted ads: some users never clicked on the ads; some others uninstalled the app immediately after they realised that it collects their personal information for advertising purposes.

To conclude, the permissions mechanism of the Android system is powerful, yet there are (at least) two significant problems for users that need to be addressed. First, many users with a lack of privacy awareness install apps without considering the permissions: a study [21] shows that only 17% of 308 Android users paid attention to permission information during the installation, and only 3% of 25 participants could correctly answer three questions pertaining to permission comprehension. Second, even if a user has carefully analysed the permissions, the model provides a 'take-it-or-leave-it' offer: the user must permanently grant all permissions in order to install the app. It may be argued that, to give users more flexible control over their personal information, a complementary mechanism would be beneficial. To this end, proposed permissions extensions for Android include PermissionTracker [22], Flow Permissions [23] and Apex [24].

An improved permissions mechanism has been announced for Android 6.0 Marshmallow, which allows users to enable or disable permissions of apps after the installation in the new system. However, there are still two outstanding issues. First, the new version of Android is not expected to support all existing Android devices. For example, the mobile device HTC Desire Eye cannot update to Android 5.0 at the time of writing (September 2015), even though Android 5.0 was published in late 2014. Some low-end mobiles cannot run Android 4.0, which was published in late 2011. The more detailed Android distribution numbers can be found in Table I, which was updated

[†]The first version of the 3D Digital Weather Clock has been available since June 2010, while the first version of AccuWeather has been available since January 2010.

Table I. Android distribution numbers for September 2015

Version	Codename	API	Distribution
2.2	Froyo	8	0.2%
2.3.3-2.3.7	Gingerbread	10	4.1%
4.0.3-4.0.4	Ice Cream Sandwich	15	3.7%
4.1.x	Jelly Bean	16	12.1%
4.2.x	Jelly Bean	17	15.2%
4.3	Jelly Bean	18	4.5%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	15.9%
5.1	Lollipop	22	5.1%

by Google on September 7, 2015 [25]. It is reasonable to believe that the upcoming Android 6.0 system can only be run on newer mobile devices. Therefore, a third-party permission framework is still necessary for a large number of devices that cannot be updated to Android 6.0. Second, the new feature of modifying permissions after installation can only be supported satisfactorily by the developed with the new Android M SDK. Although the operation of denying permission can also be performed on apps designed for older versions of Android, it may cause the modified apps to no longer function as intended, as prompted in Android 6.0. By contrast, returning fake or default values in the same format as the real ones, rather than simply denying permissions, can help avoiding this situation. Therefore, users will still need third-party frameworks to obtain more flexible control over the permissions of apps developed with older Android SDKs.

3. CHARACTERISTICS, GOALS AND REQUIREMENTS

In this section, we outline the goals and requirements for our solution. We start by discussing the characteristics of mobile advertising.

3.1. Characteristics of mobile advertising

1. **Timeliness and portability**. Due to the nature of the devices, smartphone users can be targeted with mobile advertising services anytime and anywhere. Meanwhile, the ad-networks can also learn personal information by analysing data collected during specific periods of time and at specific locations to, for example, infer the user's place of work and home location.
2. **Fragmented time effect**. Individuals often take advantage of the time available when waiting in line or travelling. *Fragmented time* can be spent on some simple transactions, such as browsing news or checking emails. A large number of apps have been developed with mobile advertising targetting fragmented time in mind.
3. **Various sensors**. Sensors such as GPS, gyroscopes and temperature sensors are built inside smartphones. Via utilisation of these sensors, interesting advertising solutions can be explored. For instance, Location Based Services (LBS) and Augmented Reality (AR) technology can be used together to deliver interactive ads for customers in special stores.
4. **Apps as advertising media**. Being limited by screen size, ads displayed in mobile browsers are not as conspicuous as those presented in PC browsers. In addition, due to one app being displayed per screen, ads inside apps are increasing in prevalence.
5. **Unique user**. Unlike, say, PCs, a mobile device is typically associated with a single user. However, the possibility of coerced access [26] (for example, parents checking their children's phones or children using their parents' phones to play games) needs to be considered.
6. **Permanent identifier**. In the PC context, cookies are the most common way of associating an identifier with a user. However, cookies can be cleared from browser memory, and cookies

of different browsers cannot interact with each other. By contrast, in the TMA context, advertisers can use device identifiers to track users — with these identifiers rarely changing.
7. **Viral marketing**. The viral marketing feature of mobile advertising can maximise the influence of ads with mobile users' initiating actions. For example, it may be that people will have less resistance to ads recommended by friends.
8. **More personal information**. Advertisers can obtain a preliminary profile of consumption capability on the basis of phone model. Users can be categorised into audience segments easily by analysing the categories of installed apps. Potential viral marketing targets can be located via address books and call logs. Users' various privacy contexts make it more effective for advertisers to perform TMA.

3.2. Goals

Motivated by the example of Section 2.2, we outline the following high-level goals for our solution.

1. **Privacy-preserving**. PPTMA should store and manage all kinds of users' privacy data inside the mobile device. It should allow mobile users to benefit from useful targeted mobile ads without there being a concern that their personal information will be leaked.
2. **Ad scanning and verifying**. PPTMA should scan apps and discover the advertisement plugins that they contain. In addition, PPTMA should monitor the behaviour of apps to determine whether they are behaving legitimately.
3. **Controlled access**. PPTMA should enable controlled access to users' personal information. It should provide fine-grained control over what kinds of data are made available to third parties, rather than applying the default permissions system provided by mobile platforms.
4. **Data management**. PPTMA should allow users to manage their personal information manually, meaning that they can add, delete or update the particular personal information, as well as save different copies.
5. **Business supporting**. PPTMA should serve as an interface for commercial organisations that would be happy to provide targeted ads without holding and controlling users' personal information.

3.3. Functional requirements

Based on the above goals, we establish the following functional requirements.

1. The system must be able to collect both dynamic and static personal information stored in mobile devices to establish and maintain profiles of users' interests.
2. Users must be able to revise their profiles manually.
3. Users must be able to grant or withdraw permissions to all apps at a global level.
4. Users must be able to adjust fine-grained permissions associated with apps.
5. Users must be able to obfuscate their real sensitive information and create fake personal information to share with ad-networks.
6. Users must be able to preset the overall privacy-sharing level to decide to what extent the coarse-grained information can be shared with ad-networks.
7. Users must be able to decide explicitly whether specific data can be shared with a particular ad-network.
8. The system must be able to pre-download an ad list from cooperative ad-networks and must be able to run targeted algorithms locally to select the most relevant ads to display.
9. Ad-networks must be able to share criteria for ad selection with the system and must be able to provide a relatively small list of candidate ads to the system on the basis of only coarse-grained information.
10. The system must be able to provide a view-and-click report to cooperative ad-networks without exposing the user's identifier.
11. The system must be able to monitor, distinguish and forbid access of personal information by malicious ad-networks and untrusted third parties.

3.4. Non-functional requirements

The following are included among the non-functional requirements.

1. **Power and bandwidth cost.** Our system must keep running in the background on mobile devices; as such, the power consumption level should be reasonable. The system is positioned between the database containing sensitive information and third-party apps to prevent them from interacting directly, so there will be additional communication time and network bandwidth consumption; the relevant cost to mobile devices should also be kept as low as possible. In addition, the performance of behavioural targeted algorithms should be optimised.
2. **Effectiveness and security of data management.** Mobile devices carry much more personal information than PCs do, and this information should be handled appropriately. The inherent complexity of the privacy-related data inside mobile devices makes it difficult to manage all kinds of different information effectively. Clearly, if we cannot assure the security of our system, it might be used by malicious parties to collect users' personal information.
3. **Accuracy of advertisement selections.** Since our objective is to develop a system that helps both consumers and advertisers, the accuracy of those displayed ads needs to be ensured. For that purpose, we need to propose effective targeting strategies to select more accurate ads for the cooperative ad-networks. (Inevitably, the accuracy will be reduced, but we aim to ensure that this reduction in accuracy is tolerable.)

4. THREAT MODEL

Many different parties and types of personal information are involved in TMA. In this section, we examine them in detail and analyse the associated threats. We start by identifying the stakeholders.

4.1. Stakeholders

There are four main kinds of participating stakeholders in our system: advertisers; content providers; ad-networks; and users. A typical workflow involving these stakeholders might be as follows.

1. Advertisers register their ads to a particular ad-network.
2. Content providers who partner with the ad-network would leave a space on their pages or apps and call relevant APIs of the ad-network to send user profiles or display ads.
3. Once a mobile user views the page or runs the app, the ad-network will notice the operation and track the user across all its partners.
4. With the user's behavioural activities and other personal information collected from all content provider partners, the ad-network can run its behavioural targeting algorithm, which determines which ad to deliver to the particular user on the current page or application.
5. If the user actually clicks on the delivered ad, the advertiser should pay the ad-network, and the ad-network will then share the payment with related content providers.

Other than the privacy-friendly ad-networks, who need to cooperate in order to operate the system as designed, most of the other stakeholders — advertisers, content providers, and general ad-networks — can take the potential adversarial roles. We assume that our adversaries would try to collect users' personal information, secretly or compulsorily, in order to perform TMA. Thus, we consider our main adversaries to be the ad-networks that are plugged inside apps and collect users' personal information based on users' interactions with the involved apps.

Another class of adversary is untrusted third parties who try to collect users' information and trade it with others. These untrusted parties also perform the collection through apps that require related permissions. Therefore, the system should monitor the behaviour of malicious apps even though they do not contain any ad-network plugins.

Although advertisers and content providers might not interact directly with our system, there is the potential for them to be unhappy (and reasonably so) if the proposed system were to influence their interests. Click-fraud (a type of fraud that occurs in pay-per-click online advertising when the attackers click on an ad — rarely by hand, often by automated scripts — without actual interest in the target of the ad's link for the purpose of earning extra income or depleting competitors' advertising budgets [27]) could be performed by advertisers to attack competitors, or performed by content providers to earn extra income. Thus, we consider people who perform click fraud as another class of adversary, and a mechanism for click fraud detecting is required.

4.2. Personal data

We now consider some of the most sensitive types of personal information stored in mobile devices. This information is the main resource that our adversaries might try to obtain. It is initially classified into two categories: dynamic information and static information. We consider each in turn.

4.2.1. Dynamic information

1. **Location information.** Compared with OTA on PCs, a significant advantage of TMA is the effective utilisation of consumers' location information. A modern smartphone, as a portable item with various sensors and network resources, is an excellent tool to record the user's location information. Mobile apps can obtain users' precise locations using GPS or network location sources such as WiFi and mobile towers. The real-time location has the potential to enable advertisers to find consumers at particular times, for example delivering an advertisement for the nearest pizza restaurant just before lunchtime. Moreover, the record of past locations has the potential to help the advertisers discover where the user lives, where the user works, and the pattern of the user's daily routine.
2. **Clickstream.** By computing the clickstream (a list of URLs belonging to ad-networks visited by a user), advertisers can obtain users' behavioural profiles for performing TMA.
3. **Operations on apps.** Since all apps are assigned to explicit categories before they are uploaded to app stores, installations and removals of apps can expose the degree of a user's interest in particular areas. To quote the Director of digital marketing firm iCrossing, Rachel Pasqua, "Apple knows what you've downloaded, how much time you spend interacting with an app and even knows what you've downloaded over time and didn't like or deleted" [28].

4.2.2. Static information

1. **Hardware information.** Ad-networks can obtain the hardware information of mobile devices via APIs provided by companies such as Google and Apple. This hardware information includes data pertaining to device type, phone model, screen size, and camera performance. By analysing this basic information, advertisers can create some initial profiles for their customers. For instance, compared with the iPhone, the iPad is more suitable for displaying larger size ads. As another example, the fact that a consumer owns an iPhone 6 Plus will have consequences as to consumption capacity. Such information may be utilised to target ads.
2. **Unique device ID.** Every mobile device is assigned an International Mobile Station Equipment Identity (IMEI) after it is produced. The IMEI, which is unique around the world and usually used to identify valid devices, can be used to support TMA. For instance, if several application developers partner with the same ad-network then they can share the specific user's information to create a more precise profile by locating the unique IMEI. In addition, even if an app is uninstalled from a given mobile device, the ad-network can still collect that user's data with other apps. If the app is installed again or a new app belongs to the ad-network is downloaded, they can continue sharing the user's data without missing anything by identifying the IMEI. Going further, sometimes the MAC address of a mobile device is also collected as the device ID. It is semi-permanent (unless the user deliberately modifies it).
3. **SIM card information.** The information stored inside the SIM card can also be obtained easily with APIs. A SIM card associated with a mobile phone contains its phone number and SIM series number. This information enables advertisers to establish a direct contact relationship with the user. In some regions, the phone number can also indicate the network provider and the network type of a mobile phone. For instance, a phone number starting with

139 in China indicates that its network type is 2G and the network provider is the company *China Mobile*, while a phone number starting with 186 in the same country indicates that it is a 3G SIM card provided by *China Unicom*.

4. **System preferences.** Most smartphones require their users to set the system preferences when they start up the device for the first time. These preferences can provide some useful information to the advertisers: the system type and version can be used to determine which peripheral device ads should be delivered. Preferred language and region indicate the language that ads should use. Moreover, settings in system accessibility may indicate that the user has disabilities; this information can, in turn, be used to deliver targeted ads.
5. **Communication information.** Communication information includes contact information, call logs, messages, and emails. Ad-networks can use contacts and call logs to perform viral marketing. They can record users' consumption information by analysing the messages sent from their banks. For example, *Now cards* — a sub-app of the *Google* app — can automatically suggest a link to track packages by analysing a recent email confirming a purchase. With relevant permissions, which are often ignored by users, communication information is exposed to third parties.
6. **Calendar events.** With users' authorizations, which are usually neglected by users and set to default values, apps can add, remove and change calendar events and send emails to guests without the user's knowledge. The capability of operating calendars makes marketing to specific people with specific themes much easier for those advertisers.
7. **Browser cookies.** Cookies on PCs enable ad-networks to track users' browsing activities across different web-sites. The cookies associated with the built-in browsers of mobile devices also record users' activity in much the same way. Together with the unique device ID, the tracking effectiveness of mobile devices can be more effective than on PCs.
8. **Installed apps.** As mentioned above, users' operations on apps carry some useful behavioural metadata. In fact, even without tracking the operations, advertisers can still create profiles of users' interests by analysing the number and categories of the apps that have already been installed on a mobile. The fact that a mobile device has one of *NBA Game Time* or *MyNBA2K15* installed is probably sufficient to indicate an interest in professional basketball.
9. **Basic personal information.** Most mobile systems provide a function for users to create their own contact card, which contains their names and contact information. This information can be collected easily by calling the relevant APIs.

5. DESIGN

At a high level, our system is implemented as a service running in the background of the mobile system with an associated application to adjust the privacy settings. The system works as a middleware instantiation on mobile platforms, positioned between the underlying database and untrusted third-party mobile applications.

5.1. Design guidelines

Haddadi *et al.* [29] describe the concept of a *databox* — a trusted platform with facilities for data management, a fine-grained access control mechanism for personal information, means for users to interact with their own data, and the ability to support innovative uses with commercial organisations. In many ways, our system is an instantiation of such a databox. In addition, we have adopted the foundational principles of *Privacy by Design* (PbD) [30], which serves as a reference framework that guides the system design with detailed criteria for privacy preservation. Altogether, the following are taken into account in our design process.

1. **Meet the requirements of a databox.** The architecture should be designed to meet key elements of a databox: not only should it provide centralised management of user privacy, but it should also enable users to interact with their profile and support legitimate uses for cooperative ad-networks.

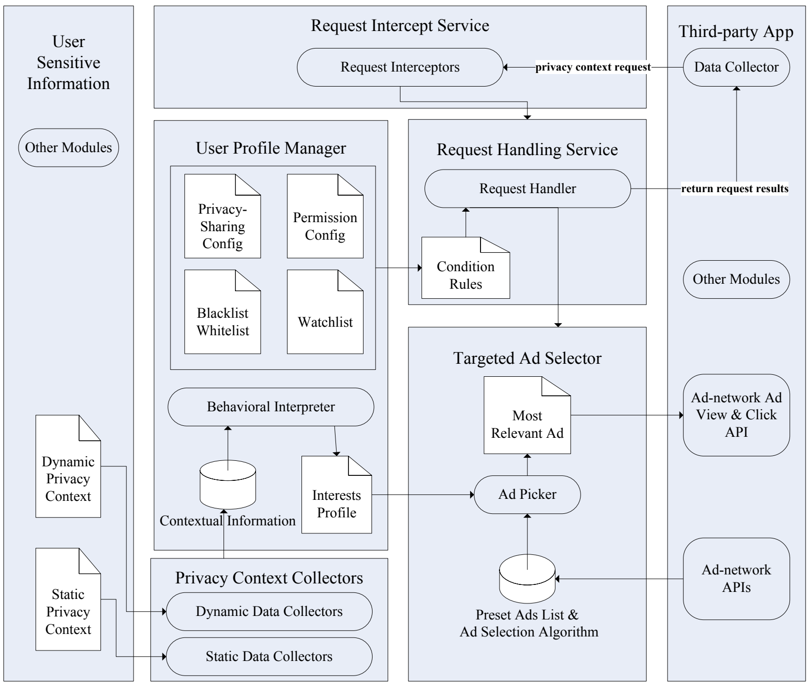


Figure 5. PPTMA architecture

2. **Compliance with the PbD principles.** The maximum degree of privacy should be set as default, and our design should pursue the objective in a positive-sum win-win manner. The design attempts to balance the relationship between the needs of ad-networks and the needs of mobile users — while we respect user privacy, our design explores a manner to achieve a positive-sum approach instead of a zero-sum one.
3. **Ease-of-use.** Our design should be simple enough and straightforward for users. It may be argued that, for those users who do not have significant concerns about privacy issues, a complicated system might reduce the likelihood that they accept the approach.

5.2. *System architecture*

Figure 5 shows the system architecture of our prototype system. There are five main components in our design: the request intercept service; the request handling service; privacy context collectors; the user profile manager; and the targeted ad selector. We consider each in turn.

1. **Request intercept service.** The request intercept service is a background service, which prevents third-party apps from collecting different privacy-related information, such as location, and from accessing the call log directly. Each time a third-party app asks for the user's personal information by calling related system APIs, the corresponding interceptor will block the request by hooking those APIs. The details of the request — such as the target privacy context, the request time, and the name and version of the third-party app — will then be delivered to the request handler.
2. **Request handling service.** By analysing details of requests and checking privacy-sharing settings, the request handler computes appropriate request results and returns them to the third-party apps. For example, if the information of the third-party app indicates that it contains the plugins of a cooperative ad-network, the request will be delivered to the targeted ad selector to provide the most relevant ad locally. If the third-party app is on the user's whitelist, the request handler will cancel the block and enable the app to call related APIs directly. If fine-grained access control of the given app was set by the user, the request handler will grant corresponding permissions by checking related settings from the profile manager.

3. **Privacy context collectors.** There are various privacy context collectors that can be classified as dynamic or static data collectors. These collectors monitor and fetch all privacy contexts (see Section 4.2) to update the contextual information database for the centralised maintenance. For example, the location information is collected by calling related APIs every 30 minutes. All location records are then saved into the contextual information database.
4. **User profile manager.** The user profile manager consists of the following parts:
   - the contextual information database, which stores various personal information collected from the mobile;
   - the behavioural interpreter, which computes the data from the contextual information database to construct the user interest profile;
   - the user interest profiles, which can be used to infer the user's interests;
   - the privacy-sharing config that describes the overall privacy sharing level;
   - the permission config that describes the fine-grained access control for each app;
   - a blacklist and a whitelist that explicitly indicate the threat level of apps; and
   - a whitelist that records those apps that need to run in a sandbox.

   The user profile is not universal across all cooperative ad-networks. Set by the user, the profile could be instantiated to different copies for each ad-network. Each copy consists of a configuration file and a set of data, either authentic or fake, that record the user's basic information (e.g. location, operations on apps, hardware, installed apps, and so on) and deduced interests. By default, the original user profile will be shared, inside the mobile device, with ad-networks that appear on the whitelist. In addition, the related files and data will be obfuscated or encrypted to protect the user profile from malicious apps.
5. **Targeted ad selector.** The targeted ad selector pre-downloads some ad lists from cooperative ad-networks and stores the ad selection algorithm that they share. When combined with the interest profile from the user profile manager, the ad selector is then able to select the most relevant ad to display.

In addition, our system imports the limited versions of the ad-SDKs of cooperative ad-networks to perform the necessary functions — pre-downloading ads by providing only limited anonymous user information and submitting view and click reports without particular user identifiers. Thus, cooperative ad-networks and related apps will have no need to call APIs of our system.

6. PROTOTYPE IMPLEMENTATION

The prototype implementation of our architecture is an Android service associated with an app. The service starts automatically after the booting of the system for the purpose of hooking sensitive APIs and monitoring malicious behaviour, while the associated app is used to adjust the related privacy configuration. The prototype consists of five functional modules:

1. The **ad scanning** module scans plugins of ad-networks from installed apps and provides support for cooperative ad-networks.
2. The **permissions management** module works as a complement to the existing permission mechanism and offers fine-grained access control over each installed app.
3. The **personal profile management** module enables users to maintain their personal information and create mock copies.
4. The **sandbox** module monitors sensitive operations performed by untrusted apps.
5. The **whitelist and blacklist** module divides installed apps to different groups — trusted or malicious apps.

6.1. Ad scanning

Users can determine whether or not an app contains ads and then configure it for further processing. The method we use to scan ads is feature library comparing. To collect personal information and

Table II. Different ad styles supported by *Domob*

Classname in domob SDK	Represented ad style
`cn.domob.android.ads.AdView`	Banner ads
`cn.domob.android.ads.FeedsAdView`	Feeds ads
`cn.domob.android.ads.InterstitialAd`	Interstitial ads
`cn.domob.android.ads.RTSplashAd`	Real time splash ads
`cn.domob.android.ads.SplashAd`	Splash ads

deliver ads with apps, developers need to register their apps with ad-networks, import their libraries, declare related permissions, and call ad-APIs provided by ad-SDKs.

The typical process of ad scanning could be briefly described as follows.

1. Call related Android APIs to get the source locations of all installed apps.
2. Get the `.dex` files (Dalvik Executable files, converted from Java `.class` files) from the source locations.
3. Traverse all classes of `.dex` files, and compare the classnames with those related to ad libraries.

If the particular classnames are found in the `.dex` files, then we can determine that the involved apps do contain ads provided by the particular ad-networks. In addition, as the permission of getting source locations of apps is allowed by default, our framework does not need any extra permissions. A typical app with ad plugins provided by the ad-network `Domob` may work as follows:

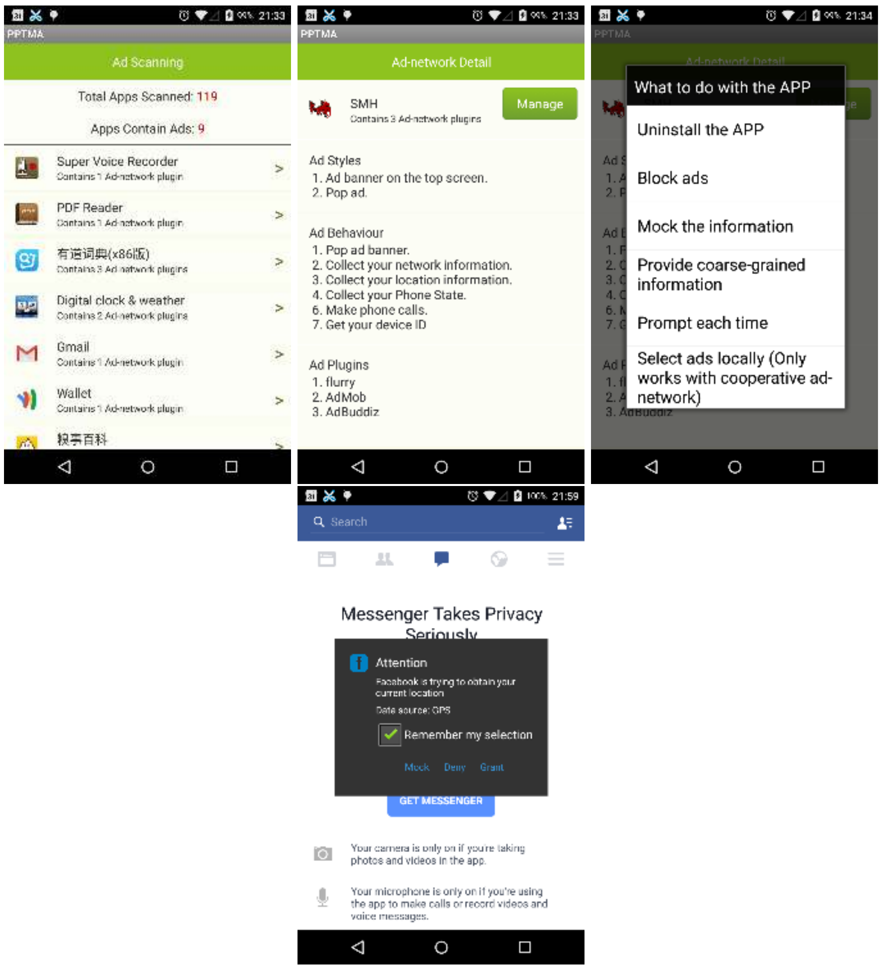
1. Register with `Domob` and get the unique publisher ID for the app.
2. Import the ad library `domob_android_sdk.jar` to the app, then declare requested permissions such as `READ_PHONE_STATE`, `WRITE_EXTERNAL_STORAGE`, `ACCESS_COARSE_LOCATION`, etc.
3. Call related ad-APIs according to different styles of ads to be displayed (see Table II).

Ad libraries are usually compiled to Java Archive (`.jar`) files, which can be detected from an installed app or its installation package. By comparing jars, classnames and used APIs in apps with typical ad libraries, we can obtain the following information of a particular app (see Figure 6):

1. The kinds of ad styles an app contains, such as banner ads or pop-up ads, which can be deduced from the called ad-APIs and related classnames (see Table II).
2. The list of ad behaviour deduced from the permissions declared in the app, such as collecting device ID or location.
3. The particular ad-networks an app contains, obtained by scanning the imported jars.

After the scanning, users can adjust the privacy strategy of involved apps to protect their personal information. Options include:

1. Uninstall the app.
2. Block ads. Users can withdraw permissions required by the ad-plugins to block ads. This strategy can also be executed by decompiling the Android Application Package (APK) or blocking related APIs.
3. Mock the information. Users can provide the preset information, which can be mocked in the personal profile management module, instead of their real information to ad-networks.
4. Provide coarse-grained information. In this strategy, only some coarse-grained information will be shared with ad-networks, such as the user's approximate location and gender.
5. Prompt each time. The system will ask the user to make a decision explicitly each time the app asks for a certain kind of personal information (see the last user interface in Figure 6).
6. Select ads locally. For apps containing cooperative ad-networks, the system can run an ad selection algorithm locally.



PPTMA

Ad Scanning

Total Apps Scanned: 119

Apps Contain Ads: 9

Super Voice Recorder
Contains 1 Ad-network plugin

PDF Reader
Contains 1 Ad-network plugin

有道词典(x86版)
Contains 3 Ad-network plugins

Digital clock & weather
Contains 2 Ad-network plugins

Gmail
Contains 1 Ad-network plugin

Wallet
Contains 1 Ad-network plugin

糗事百科

PPTMA

Ad-network Detail

SMH
Contains 3 Ad-network plugins

Manage

Ad Styles
1. Ad banner on the top screen.
2. Pop ad.

Ad Behaviour
1. Pop ad banner.
2. Collect your network information.
3. Collect your location information.
4. Collect your Phone State.
6. Make phone calls.
7. Get your device ID

Ad Plugins
1. flurry
2. AdMob
3. AdBuddiz

What to do with the APP

Uninstall the APP

Block ads

Mock the information

Provide coarse-grained information

Prompt each time

Select ads locally (Only works with cooperative ad-network)

Search

Messenger Takes Privacy Seriously

Attention

Facebook is trying to obtain your current location

Data source: GPS

Remember my selection

Mock Deny Grant

GET MESSENGER

Your camera is only on if you're taking photos and videos in the app.

Your microphone is only on if you're using the app to make calls or record videos and voice messages.

Figure 6. Ad scanning module

6.2. *Permissions and personal profile management*

To collect more personal information for TMA, many apps require extra permissions that they do not actually need for general app functions. In the permissions management module, users can browse how many permissions each app holds and revise them. By doing so, the take-it-or-leave-it approach of the current Android permissions mechanism can be addressed — users can grant all permissions for the purpose of installing an app, then draw back permissions to protect personal information.

Typically, there are three approaches to implementing such an extension: app decompiling, firmware modifying, and API hooking. We discuss each in turn.

App decompiling. Android apps are actually archive files in zip format with the filename extension `.apk`. Each APK file contains the file `AndroidManifest.xml`, which declares all permissions the app requires, and the file `classes.dex`, which holds all of the app's code. By decompiling and repacking the APK file, we can remove certain permissions from the `AndroidManifest.xml` file or insert stubs around sensitive APIs, which can be detected in the `classes.dex` file. This approach can be applied to all Android systems and devices without root permission. As an app-based approach, it enables users to delete any Android permissions an app requires without damaging the system. However, there are still disadvantages with this approach. For example, after the decompilation, users need to reinstall the app — it is inevitable that some configuration and history may be lost in this process. In addition, some pre-installed apps cannot

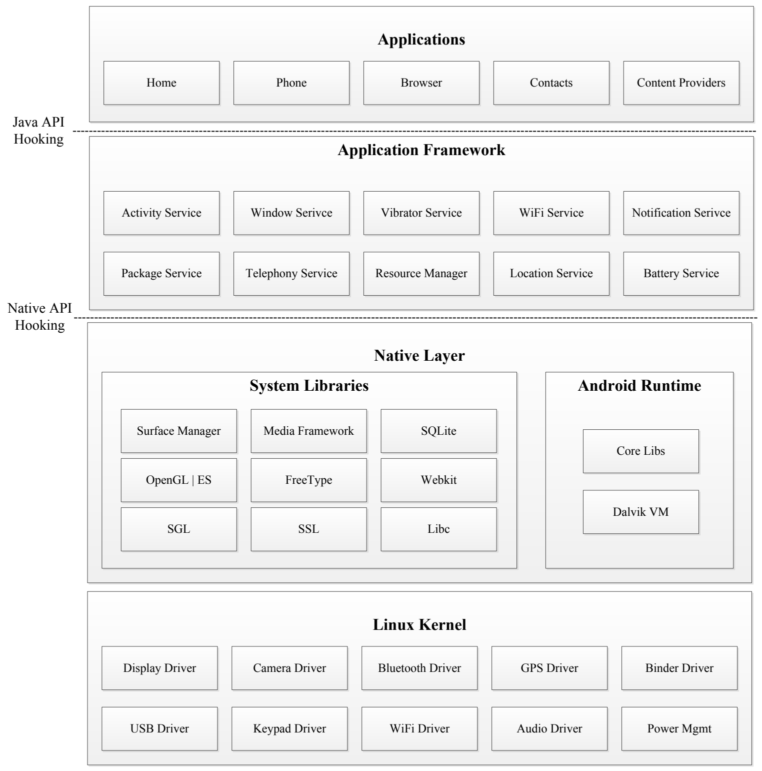


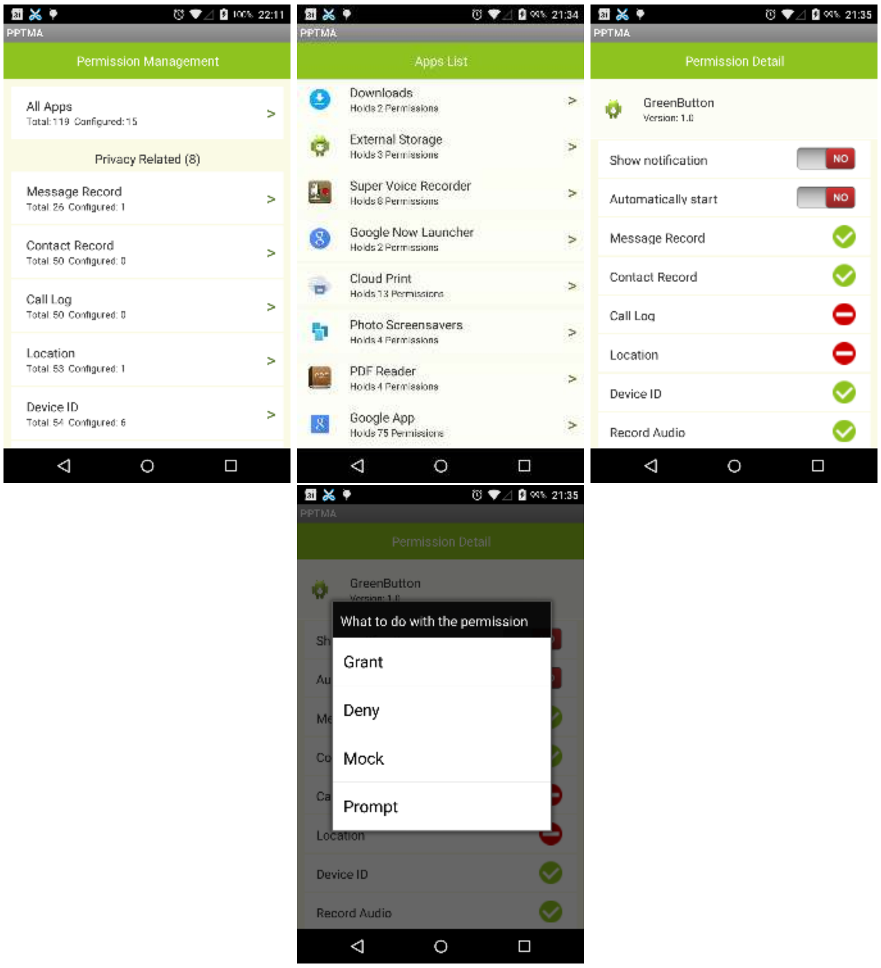
Figure 7. The Android framework (after [31])

be uninstalled or reinstalled. The digital signature of the repackaged app will be modified, so those apps with extra self-checking mechanism may not work after the repackaging. Forcibly deleting some permissions may cause 'force close' problems when the app performs operations without the related permissions declared in the `AndroidManifest.xml` file. Moreover, the modifying of APK files may even cause copyright issues.

Firmware modifying. To implement permission extensions, some researchers and developers provide patches for the existing Android system. An alternative approach involves taking advantage of the open nature of Android and modifying the source code, with a view to leveraging third-party firmware. This approach offers many benefits and allows developers to make any changes to the permissions system. However, the system patching or the firmware flashing progress is relatively complicated and unreliable: users without the relevant knowledge and skills can easily damage their system. In addition, system patching only matches particular Android systems, while firmware flashing only matches particular Android devices. Therefore the applicability and ease-of-use of this approach means that it does not meet our design guidelines.

API hooking. Apps collect users' personal information and execute permissions primarily by calling Android APIs. By hooking related APIs we can block their requests, withdraw certain permissions, and modify the return value of APIs with fake information. There are some shortcomings with this approach: it can only be applied on devices with root permission, and it requires a service to run in the background continuously to monitor callings of APIs, which may increase power consumption and impact the performance of installed apps.

Taking the above into account, as well as factors such as portability and ease-of use, we have implemented our prototype via the API hooking approach.



PPTMA

Permission Management

All Apps
Total: 119 Configured: 15

Privacy Related (8)

Message Record
Total: 26 Configured: 1

Contact Record
Total: 50 Configured: 0

Call Log
Total: 50 Configured: 0

Location
Total: 53 Configured: 1

Device ID
Total: 54 Configured: 6

PPTMA

Apps List

Downloads
Holds 2 Permissions

External Storage
Holds 3 Permissions

Super Voice Recorder
Holds 6 Permissions

Google Now Launcher
Holds 2 Permissions

Cloud Print
Holds 13 Permissions

Photo Screensavers
Holds 4 Permissions

PDF Reader
Holds 4 Permissions

Google App
Holds 75 Permissions

PPTMA

Permission Detail

GreenButton
Version: 1.0

Show notification	NO
Automatically start	NO
Message Record	✓
Contact Record	✓
Call Log	⛔
Location	⛔
Device ID	✓
Record Audio	✓

What to do with the permission

Grant

Deny

Mock

Prompt

Figure 8. Permissions management module

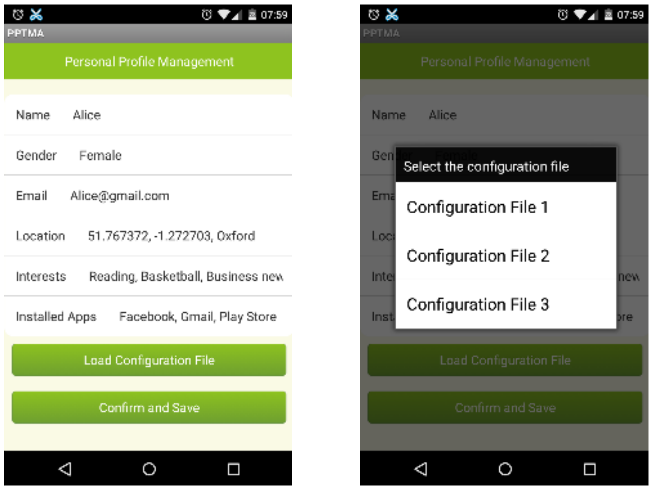
Figure 7 shows the framework of the Android platform. Much of the framework is implemented in Java, with a significant amount of code compiled to native machine code. Hence, we can execute the code injection and hook the system in two layers — the Java layer and the native layer.

While hooking in the native layer could offer more powerful functions, most apps collect personal information — location, installed apps, etc.— by calling related Java APIs. Therefore, for the purpose of intervening permission mechanism, providing return values with mock data, and running our own ads selection algorithm instead of those of the ad-networks, we mainly hook sensitive Java APIs at run-time. In the current version of our prototype, we use Cydia Substrate [32], a platform for customising software on Android, to accomplish API hooking.

The user interface of this module is shown in Figure 8. For each permission, there are four options: grant, deny, mock, and prompt.

In the Android system, denying a permission completely may sometimes result in a 'force close', therefore the prototype does not 'deny' permissions, but, rather, returns default fake values when apps call related APIs. For example, returning 'latitude = 1.0 and longitude = 1.0' for location, or 'phoneNumber = 00000000000' for the phone state.

In the personal profile management module (see Figure 9), users can preset different copies of their personal profiles, whether real (collected by privacy context collectors of PPTMA) or fake (edited by users themselves). The selected edition will be provided to the third-party apps when mocked information is required in the permissions management module or the ad scanning module.



Name	Alice
Gender	Female
Email	Alice@gmail.com
Location	51.767372, -1.272703, Oxford
Interests	Reading, Basketball, Business new
Installed Apps	Facebook, Gmail, Play Store

Figure 9. Personal profile management module

For example, a user living in Oxford can make fake copies to offer a different location: she can decide how accurate the information that she would like to share with third-party apps should be.

6.3. *Sandbox*

Users can add an unknown app, an app that may contain dangerous or malicious ads, or an app that may collect their personal information into the sandbox. The system will monitor incoming and outgoing network data, information leaks via the network, and suspect operations performed by the app. For instance, when a malicious app in the sandbox is trying to obtain the user's location, it needs to call the `android.location.LocationManager.requestLocationUpdates` API to update the user's most recent location, and it also needs to call `android.location.LocationManager.getLastKnownLocation` to get the particular information. By hooking relevant APIs, we can monitor the app's operations related to location information.

By monitoring calls of the `android.telephony.SmsManager.sendTextMessage` API, we can observe whether an app inside the sandbox is trying to send an SMS message without the user's knowledge. In addition, we can also obtain the destination, the source, and the text of the SMS message. According to different preset privacy-preserving levels, the sandbox can not only monitor the behaviour of malicious apps, but can also block their operations or even modify parameters and results involved in those operations.

The Android OS is based on the Linux kernel, and system calls of the Linux kernel can provide useful functions to apps when they perform operations on network or files. Accurate information about behaviours of apps can be obtained by capturing and analysing the system calls [33]. Other than system calls, file system logs and APIs provided by the Android system (e.g. `Android.net.TrafficStats`, which can be used to monitor the network traffic of each app) can also be used to constitute the sandbox mechanism.

Importantly, the sandbox is complemented by a blacklist. Furthermore, like similar Android application sandbox approaches (e.g. [33, 34, 35]), the system can help detecting potential misbehaviour by third-party apps. Judged by the sandbox automatically, apps with potential malicious intents are marked as 'suspect malicious apps' and reported to users for a final judgement.

With support from the sandbox mechanism, and manual input from users, installed apps can be added into the whitelist or the blacklist. Apps in the whitelist will be completely trusted and be able to obtain all information they require, while all permissions of apps in the blacklist will be withdrawn so that they can obtain only mocked information.

Table III. Software testing environment

Type of apps	Quantity	Representative
Free off-the-shelf apps	60	Facebook, Doodle Jump
System built-in apps	69	Gmail, Package Access Helper
Apps for daily use	59	BBC iPlayer Radio, OfficeSuite Pro 8
Dummy apps	10	DummyBanner, DummyInterstitial
PPTMA client	1	PPTMA
API hooking auxiliary	1	Cydia Substrate

Table IV. Cost of the prototype

Term	Average	Sample variance
Memory consumption	9066KB	4855471 (n=10)
Power consumption in standby mode for 24 hours	less than 1%	N/A
Ad scanning time per app	24.82 ms	799.23 (n=200)
Time of obtaining real location	5.19 ms	124.39 (n=100)
Time of obtaining and faking location	5.91 ms	114.97 (n=100)
Time of sending a SMS message without monitoring	4.05 ms	5.20 (n=100)
Time of sending a SMS message and recording details	4.81 ms	6.91 (n=100)
Time of loading original ad url from dummy ad-network	0.01 ms	0.009 (n=100)
Time of replacing original url and loading new local ad url	0.02 ms	0.019 (n=100)

6.4. Evaluation

Our test device for our prototype is a LG Nexus 4 (1.5 GHz quad-core Krait, 2 GB of LPDDR2 RAM clocked at 533MHz), which runs a rooted firmware based on Android 4.3.1. The software testing environment consists of 200 apps in total: 60 off-the-shelf apps selected randomly from the 150 top free apps of the Google Play Store as of September 2015; 69 built-in apps that come with the firmware (some of the apps can be operated by users, for example, Gmail and Google Map; the others, such as Themes Provider and Calendar Storage, are hidden from users); 59 apps for daily use, including both free and paid apps; 10 dummy apps that consume ads from testing cooperative ad-networks built to work with our system; one app as the PPTMA client that can be used to adjust related configurations; and one app as the client of Cydia Substrate that helps to accomplish API hooking. The details are shown in Table III.

According to the Android memory management mechanism, if the system runs low on memory, processes may be killed in the LRU cache beginning with the processes least recently used. In order to keep a stable testing environment, we only select three apps from each of the first four types of apps to run, with one dummy app which consumes banner ads from the dummy ad-network running in the foreground. Each dummy app tries to get a banner ad from the dummy ad-network after it starts. It also tries to get the location from the device and send an SMS message to the ad-network without telling the user in every second. When the PPTMA system works, APIs related to obtaining location and sending SMS messages are hooked: fake locations are submitted to the apps; operations of sending SMS messages are monitored; and details such as the destination, the source, the text content, the date, and the process ID of the app are recorded.

We first evaluate the costs of the prototype in terms of the memory and power consumption, then observe its impact on the performance of the third-party apps, and, finally, we evaluate the efficiency of ad-scanning. We measured time costs of API hooking and ads replacing by averaging over several runs of one dummy app which contains job timer in its code. Test data is shown in Table IV.

Table V. Compatibility testing

Term	Total number of apps	Passing number of apps	Passing rate
Location hooking	46	46	100%
SMS hooking	15	13	86.7%

As expected, the costs of memory and power consumption are relatively low compared to the totality of the available resources. In addition, there is a performance overhead of only 13.9% (from 5.19 milliseconds to 5.91 milliseconds) after the location API hooking on third-party apps, which is also acceptable. The overhead pertains to stopping apps from getting real data from GPS sensors and replacing the return value with fake data. A performance overhead of 32.9% occurs after monitoring and recording the operations of sending SMS messages. Although 32.9% seems relatively high, the base number is actually very small (4.05 milliseconds).

The time consumption of ad scanning is currently quite high: it costs about 5 seconds to scan 200 apps. However, since the full scanning only happens when users scan their apps for the first time, we would argue that the waiting time is tolerable. The storage sizes of sample apps range from 0.53MB to 144MB, which contribute to the high sample variance. In addition, an incidental result shows that at least one ad-network plugin was found in 46 out of the 60 apps selected from the top free apps of the Google Play Store.

We also utilise the dummy apps to test the function of replacing original ads obtained from ad-networks with new ads selected from the local ads pool inside the mobile device. The result suggests that the information of ads can be changed and recorded successfully by PPTMA once we get the method names of loading ads. The progress can become much easier if we can obtain the limited versions of the SDKs from cooperative ad-networks — so that we do not need to spend time on detecting, hooking, and modifying their ad APIs. To clarify, time costs of loading and replacing ads mentioned in Table IV only pertain to the operation of modifying the url of targeted ads. Since the prototype is only a starting point for us to develop and refine local ad selection algorithms, evaluation of selecting the most relevant local ad for users is not an issue at this time.

The prototype hooks location-related and SMS-related APIs. The former validates the functions of permission and personal profile management, while the latter validates the sandbox mechanism. Testing on app compatibility of the API hooking mechanism was performed, with the test data shown in Table V.

The location hooking mechanism function properly in all of the 46 apps that require permissions for obtaining location, including some popular apps such as Facebook, Instagram and Google Maps. In terms of the SMS hooking mechanism, 15 apps hold the permission of sending SMS messages. However, in two of them we could not find a related interface to perform the sending operation. Monitored by our system, the two apps did not send SMS messages in the background. Although it is likely that they over-claimed the permission of sending SMS messages, we assume that they can send SMS messages silently and avoid our monitor because it is not obvious how to trigger the SMS sending progress in these apps. The SMS hooking mechanism works well in the remaining, with a passing rate of 86.7%.

We must acknowledge that with more APIs being hooked by our framework, as well as more cooperative ad-networks being imported to it, there might be a corresponding increase in the consumption of memory and power. In addition, the performance of hooking APIs and obfuscating information might also be affected. However, since these new features are designed as individual add-ons for our main framework, and the system resources are mainly consumed by the framework, the influence of new add-ons on consumption and performance is expected to be tolerable.

6.5. Summary

The prototype implementation meets our basic requirements and design guidelines of our architecture, as discussed below. Users can preserve their personal information in different levels

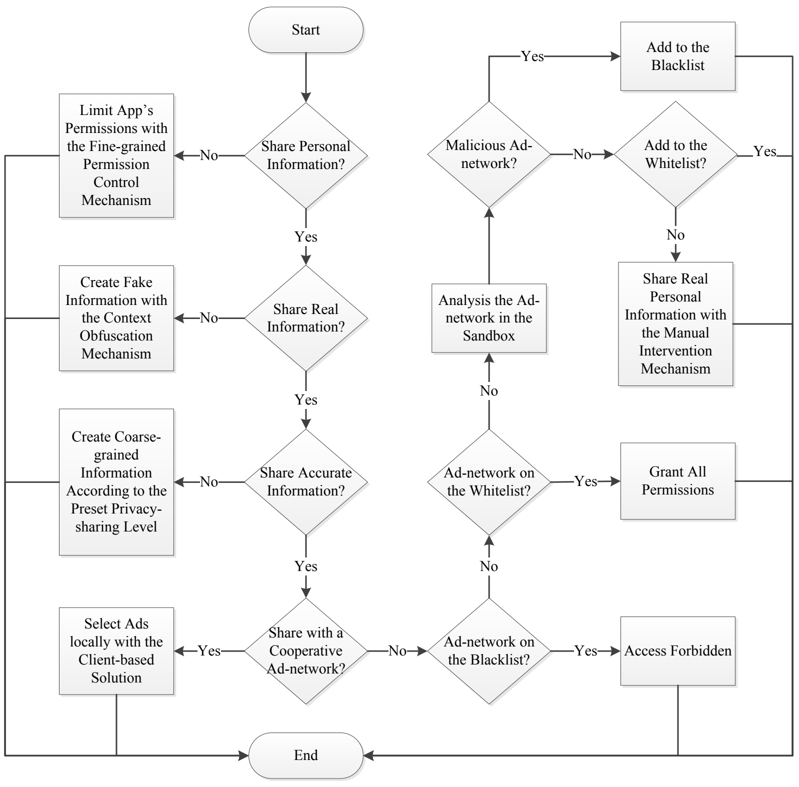

Start
Share Personal Information?
Limit App's Permissions with the Fine-grained Permission Control Mechanism
No
Yes
Share Real Information?
Create Fake Information with the Context Obfuscation Mechanism
No
Yes
Share Accurate Information?
Create Coarse-grained Information According to the Preset Privacy-sharing Level
No
Yes
Share with a Cooperative Ad-network?
Select Ads locally with the Client-based Solution
Yes
No
Ad-network on the Blacklist?
Yes
Access Forbidden
No
Ad-network on the Whitelist?
Yes
Grant All Permissions
No
Analysis the Ad-network in the Sandbox
Malicious Ad-network?
Yes
Add to the Blacklist
No
Add to the Whitelist?
Yes
No
Share Real Personal Information with the Manual Intervention Mechanism
End

Figure 10. Personal information process flow

and retain useful services of third-party apps with the prototype. A related flow chart of privacy context handling strategies is shown in Figure 10.

1. At a high level, a user can decide whether or not to share their personal information with particular ad-networks. If they choose not to share anything, they can withdraw all permissions relevant to their personal information from those apps. These functions are supported by the fine-grained permissions control mechanism in the permissions management module.
2. If the user chooses to share their personal information, no matter real or fake, to enable third-party apps to run in the way they are designed, they can decide whether or not to share real data. A context obfuscation mechanism in the personal profile management module makes it possible for the user to create and submit fake information.
3. If the user chooses to share real information, then they can share real but coarse-grained information — for example, network-based approximate location instead of GPS and network-based precise location. This feature is supported by the permissions management module and the personal profile management module.
4. If the user chooses to share accurate information and the involved ad-network is one of those willing to gain consumers' trust and to work with our architecture, the sensitive personal information will be kept inside the mobile and associated with selected relevant ads locally.
5. If the given ad-network is not a cooperative partner, then it will be checked with the blacklist and the whitelist mechanism or monitored in a sandbox. For those apps not associated with these mechanisms, a manual intervention mechanism can be applied to them. For example, an explicit prompt will appear to ask the user whether the specific data could be shared with the ad-network when personal information is requested.

7. RELATED WORK

Adnostic [2] allows PC users to provide only coarse-grained information about themselves while detailed personal profiles are used to select relevant ads and kept locally. Its prototype is implemented as a Firefox extension with two modules: the user profiling module builds a list of user interests inside the browser and the ad rendering module selects ads to be display locally. The other feature of Adnostic is that its billing system can report which advertisement was clicked without exposing this to the ad-network, so that the ad-networks can charge advertisers for clicks or impressions in a privacy-preserving way. This feature is supported by using zero-knowledge proofs and homomorphic encryption. The number of views for each advertisement are sent to a trusted third party to decrypt them first, and then returned back to the ad-network. However, there exists a significant limitation of Adnostic: it can work only with the cooperation of ad-networks. Ad-networks need to modify the way they serve ads in order to support Adnostic.

The Privad project [36, 37] has a similar goal to Adnostic. Privad contains four entities: client, dealer, monitor and broker. In this model the system downloads all potential ads from the broker, and selects the appropriate ads on the client. The dealer works as an anonymous proxy between the client and the broker, so that the broker cannot identify the client. Moreover, communications between the broker and the client are encrypted with a public key mechanism, therefore even the dealer cannot fetch the information about which ads have been downloaded. The monitor is used to ensure that the client cannot send information to the broker through a covert channel. In addition, the Privad project team proposed a manual intervention mechanism, which uses subscription-based prefetching to enable users accessing the relevant ads whenever a proxy is available. To realise that, users need to manually subscribe to the categories of ads that they are interested in before they start browsing. Unlike Adnostic, Privad does not trust ad-networks at all and it anonymises all of the information sent by the client. Consequently, the anonymising operations during its function would impact performance and consume more resources to detect click-fraud. Since the pay-per-click model is so popular, the approach adopted by Privad is unlikely to find favour with advertisers.

MobiAd [26, 38] is a private advertising system for mobile platforms. Ads are selected locally from the pool of ads, and only those ads related to users' interests will be downloaded, just as Adnostic has done on PC browsers. The information for user analysis such as ad views and clicks are encrypted and sent to the ad-network via intermittent Wi-Fi hotspots and other phones. Therefore the advertiser can only receive aggregate information and cannot distinguish users from each other. MobiAd has the same drawback as Adnostic: the system needs support from the ad-networks.

RePriv [39] is a policy architecture that allows users to control the extent to which their data is shared with the ad-network. In this model, the browsers can mine users' personalised information locally and create related interest profiles. Online service suppliers can register their own miners to extract the additional information from users' operations. Each time they ask for the user's data, an explicit prompt asks the user whether specific data from the interest profile can be submitted to the supplier. Therefore, users have complete control over their personal information, as long as they can tolerate the regular prompts.

Götz and Nath [40] proposed a framework targeted at mobile advertising with a manual intervention mechanism. Users can decide how much of their contextual information they are willing to share with the ad-network. Then the ad-network selects a list of ads for the client by computing the limited information submitted by the client. Finally the client can pick the most relevant one from the list of ads based on all information about their private context.

ProfileGuard [41] is an app-based obfuscation mechanism that works on mobile platforms that prevent third-party apps from inferring user interests by analysing installed apps. One implementation of this is in terms of an Android app. Users can run a customised ProfileGuard app and select their private interest category to protect the specific aspects of their profile. This app can analyse the information of those apps which have already been installed on the mobile and suggest the potential threat with candidate obfuscating apps to the user. The user could then select obfuscating apps to install and run so that the ad-networks cannot determine the user's real interests. The list of obfuscating apps is generated under an obfuscation strategy. The first such strategy is based on the most similar apps from the user's non-private app category, while another is customised to the user's profile interests.

With respect to the Android permissions system, many platform-level approaches have been proposed. MockDroid [42] is a modified version of the Android system, which allows users to replace sensitive personal information with fake data. TISSA [43], PermissionTracking [22], Flow Permissions [23] and Apex [24] extend the Android framework to provide fine-grained access control of specific resources or permissions.

TaintDroid [44] is an information-flow tracking system for the Android platform. It monitors untrusted apps and can be used to analyse how those apps access and manipulate users' sensitive personal information. DroidBox [34] is an Android application sandbox that uses TaintDroid to detect the leaking of information. It can provide a timeline view of the monitored app's behaviour for users to identify malicious apps. In addition, Android sandbox architectures include [45] and [46].

8. CONCLUSIONS

8.1. Possible objections

A number of questions come to mind with respect to this approach.

1. **View/click report processing.** Ad-networks, advertisers, and content providers need the click records of ads for charging money and sharing payment. Although we can keep users' personal data inside the mobile device by applying a local ad selection mechanism, we still need to download the selected ad and provide the click report, which, in turn, can be analysed by the ad-network to infer the user's interests. To provide the click records without exposing which user performs the particular operation, we propose the application of the Trustworthy Remote Entities (TRE) [47] architecture. The TRE is a computational and communication system that performs online processing of information provided by two or more communicating parties. In our proposed framework, the click records of individual users are first delivered to a TRE; the TRE then hides user IDs from different records and delivers the result to ad-networks with single-use random IDs.
2. **Click fraud.** In current TMA systems, ad-networks need the ad click reports of consumers to defend against click fraud. In our model, only single-use IDs that are randomly generated by TREs will be submitted to ad-networks with click reports, which limits the ability of ad-networks to detect click fraud. However, the mappings between the submitted IDs and real user IDs are stored in the TRE. Thus, click fraud detecting can also be performed by TREs. TREs can help to analyse suspicious records such as a single user clicks on the same ads for a number of times in a row, or different users click on a single ad in the same period of time, while the ad-networks can analyse other suspicious aspects such as the abnormal conversion rates of particular ads.
3. **Motivating advertisers.** There are a number of ways to motivate advertisers. First, since the information is edited by users themselves, rather than collected passively by unknown third parties, users are more likely to submit authentic data. In addition, as users know that the information is specifically used for selecting useful ads and the personal information is safe inside their mobile devices, they are more likely to submit accurate data voluntarily. Second, many organisations or governments have published policies to limit the ability of advertisers to track users. However, with our framework, advertisers will not need to track users. Third, since no personal information will be leaked when applying our algorithm, users will reduce the hostility towards advertisers, which, in turn, could increase response rates and transactions.
4. **Motivating users.** Though our system aims to provide balance, a potential problem is users ignoring most of the functions, and submitting fake data and blocking ads. Even if ads are displayed normally, users may still not read them if they have little interest [48] or if they have serious privacy concerns [17]. Furthermore, privacy advocates also have the potential to put an end to advertising models [49]. On the other hand, Barutçu [50] suggests that users are more likely to have positive attitudes to mobile advertising if they are price-conscious or more

involved. A study conducted for Nokia by HPI Research [51] further identified key factors contributing to the acceptance of mobile advertising, which includes independent choice and mutual benefit. Findings from other studies (e.g. [52]) also suggest that consumers would be more open to receiving ads if they can obtain recognisable benefit such as special offers or discounts, and a relatively high tolerance to targeted ads can be achieved if the information of ads is perceived as useful.

5. **Business audiences.** Large companies will not give up their own ad-networks and apply our proposed solution. However, individual advertisers and small ad-networks who hold only limited resources could make use of a solution such as this.
6. **Security issues caused by root permission.** Our framework requires rooted devices. It is possible that more information could be stolen by malicious apps because of the root permission. To address this, we provide a compensation mechanism — a sandbox to monitor the behaviour of third-party apps. The sandbox could detect and track the information leakage caused by malicious apps, with which mechanism users can take action in time. Compared with other methods, such as app decompiling or system modifying, our choice of rooting device and hooking API are the most simple methods for users to apply privacy-preserving tools. Nevertheless, we will keep exploring the possibility of applying our solution without root permission.
7. **Ad blocking.** The fact that mobile users could make use of our solution to block ads from particular ad-networks or apps would make advertisers unhappy. However, even in the settings for Google Ads, users can choose to opt-out of interest-based ads on Google, and opt-out of interest-based ads across the web. Apple also provides an ad-blocking feature in iOS9 for users to block ads on their mobile devices. To establish a healthy environment, we should give users the choice to opt out of the personalised experience if they would like to. This could, in turn, drive advertisers to improve the quality of their ads.
8. **Permissions dilemma.** If a weather app requires location details both for the local weather and for targeted ads, users may struggle with the relevant permission. In most cases, location information cannot be used alone to recognise a user: third-party apps need to collect the user's information, together with an identifier such as the device ID. Otherwise they may need to assemble many different attributes, such as mobile module, system version, location and feature of installed apps to verify a user. Thus, we can offer suggestions such as 'never submit Device ID' or 'do not submit a particular combination of information'.

8.2. Final thoughts

We have described the PPTMA architecture, which we have designed to be a consumer-targeted but business-palatable approach that helps consumers take advantage of mobile ads without compromising their privacy. We have described a prototype implementation based on the Android system that can be dynamically configured to hook sensitive APIs at run-time to enable mobile users to take control over their sensitive information.

Our solution provides a trusted platform that offers users a centralised management approach for users' personal information and provides a fine-grained access control mechanism for controlling installed apps. The system also support innovative uses with cooperative ad-networks. However, our proposed system also has the same challenges as faced by a databox [29]: availability, data complexity, cost, and so on. In addition, we must acknowledge that we cannot protect all personal information, even if we were to combine all existing privacy-preserving solutions: we cannot prevent Apple's algorithms from knowing which apps users have downloaded and installed; we cannot stop Google's algorithms from analysing emails in Gmail and calendar events in Google Calendar. It is unarguable that, in order to preserve consumers' privacy appropriately, technical and legal solutions, as well as education, also have a role to play.

As we have seen, although there is a great deal of sensitive information stored in mobile devices, many users of these devices have a relatively low level of privacy awareness. Others may claim that they understand the privacy risks, but, in fact, they fall victim to the privacy paradox. Users who really care about their personal information might uninstall the TMA apps immediately and find other alternatives. People have become accustomed to this situation, so why would they choose to adopt and deploy a novel solution?

Users clearly have an incentive to install apps that could provide financial benefits for them. Some reports indicate that mobile users search daily for money saving vouchers and local promotions with apps [26]. Privacy-aware users can enjoy financial benefits without their privacy being compromised; users with a low level of privacy awareness can treat the solution as a compensation mechanism. The incentive for ad-networks could be obtaining the same or better advertising effectiveness while letting users take control of their own sensitive information to reduce hostility. In the long term, this has the potential to engender a healthier balance between ad-networks and consumers.

Our prototype implementation provides us with a starting point to start to develop and evaluate algorithms for selecting ads on the basis of user preferences and permissions. This will be the immediate focus of our future work. In addition, we will undertaking further testing and development of the prototype.

ACKNOWLEDGEMENTS

The authors thank the reviewers for their helpful feedback. Yang Liu acknowledges the scholarship from China Scholarship Council (File No.201508060193).

REFERENCES

1. Cole AM. Internet advertising after Sorrell v. IMS Health: A discussion on data privacy & the First Amendment. *Cardozo Arts & Entertainment Law Journal* 2012; **30**:283–315.
2. Toubiana V, Narayanan A, Boneh D, Nissenbaum H, Barocas S. Adnostic: Privacy preserving targeted advertising. *Proceedings of the 17th Annual Network and Distributed System Security Symposium (NDSS 2010)*, 2010.
3. Samu S, Krishnan HS, Smith RE. Using advertising alliances for new product introduction: Interactions between product complementarity and promotional strategies. *The Journal of Marketing* 1999; **63**(1):57–74.
4. Google. Google AdSense. http://www.google.com/adsense/. [Last accessed April 2015].
5. Tsang MM, Ho SC, Liang TP. Consumer attitudes toward mobile advertising: An empirical study. *International Journal of Electronic Commerce* 2004; **8**(3):65–78.
6. Merisavo M, Kajalo S, Karjaluoto H, Virtanen V, Salmenkivi S, Raulas M, Leppäniemi M. An empirical study of the drivers of consumer acceptance of mobile advertising. *Journal of Interactive Advertising* 2007; **7**(2):41–50.
7. Awad NF, Krishnan M. The personalization privacy paradox: An empirical evaluation of information transparency and the willingness to be profiled online for personalization. *MIS Quarterly* 2006; **30**(1):13–28.
8. Barnes SB. A privacy paradox: Social networking in the United States. *First Monday* 2006; **11**(9).
9. Norberg PA, Horne DR, Horne DA. The privacy paradox: Personal information disclosure intentions versus behaviors. *Journal of Consumer Affairs* 2007; **41**(1):100–126.
10. Utz S, Krämer N. The privacy paradox on social network sites revisited: The role of individual characteristics and group norms. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace* 2009; **3**(2):article 2.
11. Debatin B, Lovejoy JP, Horn AK, Hughes BN. Facebook and online privacy: Attitudes, behaviors, and unintended consequences. *Journal of Computer-Mediated Communication* 2009; **15**(1):83–108.
12. Beresford AR, Kübler D, Preibusch S. Unwillingness to pay for privacy: A field experiment. *Economics Letters* 2012; **117**(1):25–27.
13. Martin K. Privacy notices as Tabula Rasa: An empirical investigation into how complying with a privacy notice is related to meeting privacy expectations online. *Journal of Public Policy & Marketing* 2014; .
14. Calo R. Against notice skepticism in privacy (and elsewhere). *Notre Dame Law Review* 2013; **87**:1027–1073.
15. Milne GR, Culnan MJ. Strategies for reducing online privacy risks: Why consumers read (or don't read) online privacy notices. *Journal of Interactive Marketing* 2004; **18**(3):15–29.
16. Tucker CE. The economics of advertising and privacy. *International Journal of Industrial Organization* 2012; **30**(3):326–329.
17. Tucker CE. Social networks, personalized advertising and privacy controls. *Journal of Marketing Research* 2014; **51**(5):546–562.
18. Marwick AE. "I'm a lot more interesting than a Friendster profile": Identity presentation, authenticity, and power in social networking services. *Association of Internet Researchers* 2005; **6**.
19. Tufekci Z. Can you see me now? Audience and disclosure regulation in online social network sites. *Bulletin of Science, Technology & Society* 2008; **28**(1):20–36.
20. Madden M, Lenhart A, Cortesi S, Gasser U, Duggan M, Smith A, Beaton M. Teens, social media, and privacy. http://www.pewinternet.org/2013/05/21/teens-social-media-and-privacy/. [Last accessed April 2015].
21. Felt AP, Ha E, Egelman S, Haney A, Chin E, Wagner D. Android permissions: User attention, comprehension, and behavior. *Proceedings of the 18th Symposium on Usable Privacy and Security (SOUPS 2012)*, ACM, 2012; 3.

22. Kern M, Sametinger J. Permission tracking in Android. *Proceedings of the 6th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2012)*, 2012; 148–155.
23. Holavanalli S, Manuel D, Nanjundaswamy V, Rosenberg B, Shen F, Ko SY, Ziarek L. Flow permissions for Android. *Proceedings of the 28th IEEE/ACM International Conference on Automated Software Engineering (ASE 2013)*, IEEE, 2013; 652–657.
24. Nauman M, Khan S, Zhang X. Apex: Extending Android permission model and enforcement with user-defined runtime constraints. *Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS 2010)*, ACM, 2010; 328–332.
25. Google. Android platform versions. https://developer.android.com/about/dashboards/index.html. [Last accessed September 2015].
26. Haddadi H, Hui P, Brown I. MobiAd: Private and scalable mobile advertising. *Proceedings of the 5th ACM International Workshop on Mobility in the Evolving Internet Architecture (MobiArch 2010)*, ACM, 2010; 33–38.
27. Zhang L, Guan Y. Detecting click fraud in pay-per-click streams of online advertising networks. *The 28th International Conference on Distributed Computing Systems (ICDCS 2008)*, IEEE, 2008; 77–84.
28. Myslewski, Rik. Apple ads to target your iTunes history. http://www.theregister.co.uk/Print/2010/07/06/apple_targets_ads/. [Last accessed April 2015].
29. Haddadi H, Howard H, Chaudhry A, Crowcroft J, Madhavapeddy A, Mortier R. Personal data: Thinking inside the box. *arXiv preprint arXiv:1501.04737* 2015; .
30. Cavoukian A. Privacy by design: The 7 foundational principles. *Information and Privacy Commissioner of Ontario, Canada* 2009; .
31. Chen J. An introduction to Android. http://androidgroup.googlecode.com/files/Introduction%20to%20Android.pdf 2008. [Last accessed April 2015].
32. SaurikIT LLC. Cydia substrate. http://www.cydiasubstrate.com/. [Last accessed September 2015].
33. Burguera I, Zurutuza U, Nadjm-Tehrani S. Crowdroid: Behavior-based malware detection system for Android. *Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*, ACM, 2011; 15–26.
34. Desnos A, Lantz P. Droidbox: An Android application sandbox for dynamic analysis. http://www.honeynet.org/gsoc2011/slot5. [Last accessed April 2015].
35. Jang Jw, Yun J, Woo J, Kim HK. Andro-profiler: anti-malware system based on behavior profiling of mobile malware. *Proceedings of the companion publication of the 23rd international conference on World wide web companion*, International World Wide Web Conferences Steering Committee, 2014; 737–738.
36. Guha S, Reznichenko A, Tang K, Haddadi H, Francis P. Serving ads from localhost for performance, privacy, and profit. *Proceedings of Hot Topics in Networking (HotNets 2009)*, 2009.
37. Guha S, Cheng B, Francis P. Privad: Practical privacy in online advertising. *Proceedings of the 8th USENIX Conference on Networked Systems Design and Implementation (NSDI 2011)*, 2011.
38. Haddadi H, Hui P, Henderson T, Brown I. Targeted advertising on the handset: Privacy and security challenges. *Pervasive Advertising*. Springer, 2011; 119–137.
39. Fredrikson M, Livshits B. Repriv: Re-imagining content personalization and in-browser privacy. *Proceedings of IEEE Symposium on Security and Privacy (SP 2011)*, 2011; 131–146.
40. Götz M, Nath S. Privacy-aware personalization for mobile advertising. *No. MSR-TR-2011-92, Tech. Rep* 2011; .
41. Ullah I, Boreli R, Kanhere SS, Chawla S. Profileguard: Privacy preserving obfuscation for mobile user profiles. *Proceedings of the 13th Workshop on Privacy in the Electronic Society (WPES 2014)*, ACM, 2014; 83–92.
42. Beresford AR, Rice A, Skehin N, Sohan R. MockDroid: Trading privacy for application functionality on smartphones. *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications (HotMobile 2011)*, ACM, 2011; 49–54.
43. Zhou Y, Zhang X, Jiang X, Freeh VW. Taming information-stealing smartphone applications (on Android). *Trust and Trustworthy Computing*. Springer, 2011; 93–107.
44. Enck W, Gilbert P, Han S, Tendulkar V, Chun BG, Cox LP, Jung J, McDaniel P, Sheth AN. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Transactions on Computer Systems (TOCS)* 2014; **32**(2):5.
45. Yan LK, Yin H. DroidScope: Seamlessly reconstructing the OS and dalvik semantic views for dynamic Android malware analysis. *Proceedings of the 21th USENIX Security Symposium (USENIX Security 2012)*, 2012; 569–584.
46. Spreitzenbarth M, Freiling F, Echtler F, Schreck T, Hoffmann J. Mobile-sandbox: Having a deeper look into Android applications. *Proceedings of the 28th Annual ACM Symposium on Applied Computing (SAC 2013)*, ACM, 2013; 1808–1815.
47. Paverd A, Martin AP, Brown I. Privacy-enhanced bi-directional communication in the smart grid using trusted computing. *Proceedings of the 5th IEEE International Conference on Smart Grid Communications (SmartGridComm 2014)*, IEEE, 2014; 872–877.
48. Hwang WH, Chen YS, Jiang TM. Personalized internet advertisement recommendation service based on keyword similarity. *Proceedings of the 39th IEEE Annual Computer Software and Applications Conference (COMPSAC 2015)*, vol. 1, IEEE, 2015; 29–33.
49. Jesdanun A. Ad targeting based on ISP tracking now in doubt. http://www.mercurynews.com/business/ci_10337964 2008. [Last accessed October 2015].
50. Barutçu S. Attitudes towards mobile marketing tools: A study of Turkish consumers. *Journal of Targeting, Measurement and Analysis for Marketing* 2007; **16**(1):26–38.
51. Nokia. New Nokia research shows consumers ready for m-marketing via mobile handsets. http://company.nokia.com/en/news/press-releases/2002/01/30/new-nokia-research-shows-consumers-ready-for-m-marketing-via-mobile-handsets. [Last accessed September 2015].
52. Wang K, Chen SH, Chang HL. The effects of forced ad exposure on the web. *Journal of Informatics & Electronics* 2008; **13**(1):27–38.