Don't kill my ads! Balancing Privacy in an Ad-Supported Mobile Application Market

Ilias Leontiadis*, Christos Efstratiou*, Marco Picone*†, Cecilia Mascolo*
* Computer Laboratory, University of Cambridge, Cambridge, UK
† Department of Information Engineering, University of Parma, Italy

ABSTRACT

Application markets have revolutionized the software download model of mobile phones: third-party application developers offer software on the market that users can effortlessly install on their phones. This great step forward, however, also imposes some threats to user privacy: applications often ask for permissions that reveal private information such as the user's location, contacts and messages. While some mechanisms to prevent leaks of user privacy to applications have been proposed by the research community, these solutions fail to consider that application markets are primarily driven by advertisements that rely on accurately profiling the user. In this paper we take into account that there are two parties with conflicting interests: the user, interested in maintaining their privacy and the developer who would like to maximize their advertisement revenue through user profiling. We have conducted an extensive analysis of more than 250,000 applications in the Android market. Our results indicate that the current privacy protection mechanisms are not effective as developers and advert companies are not deterred. Therefore, we designed and implemented a market-aware privacy protection framework that aims to achieve an equilibrium between the developer's revenue and the user's privacy. The proposed framework is based on the establishment of a feedback control loop that adjusts the level of privacy protection on mobile phones, in response to advertisement generated revenue.

1. INTRODUCTION

Mobile phones have become a ubiquitous piece of technology that is carried by virtually every individual throughout their daily life. The improved capabilities of smart phones (computation, sensing and communication) have transformed them into *avatars* of the individual in the digital world. Indeed, smart phones are gate keepers of one's mobility patterns, contact details of friends, social networks, etc. The combined information that can be accessed through a smart phone is vast, rich in detail, and covers a variety of the owner's personal life. At the same time, the proliferation of smart phones can be largely attributed to their ability to host a range of third-party applications that can be downloaded and installed by the user. Allowing third-party applications to operate within a device holding private information about their owner can lead to unanticipated privacy and security risks: according to [10], malicious Android applications have grown 5 times from January 2011 to July 2011. The literature includes a variety of recommended solutions for privacy protection: delivering mock information to the application [2, 12] or hiding details using differential privacy [14].

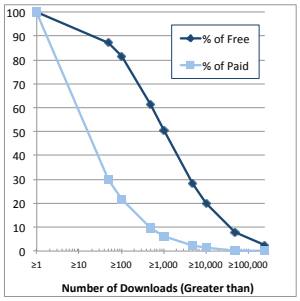
While the issues around privacy are becoming more evident over time, we also see a booming market of mobile phone applications that, to a significant extent, are free. In July 2011 Google announced that there are more than 250,000 applications (up from 5,000 applications less than two years ago) in the Android market that were downloaded more than 8 billion times by more than 100 million Android devices [13]. The mobile advertising industry is an integral part of this market, as it offers the financial incentives for developers to distribute free applications. However, the success of the advertising industry is interlinked with the accurate profiling of users who are the recipients of targeted advertisement [6]. This means that a successful advertisement campaign requires access to personal information that can potentially be considered private.

Although existing solutions to mobile phone privacy can offer some level of protection to the user [2, 12], *they fail to consider the implications for a market that is primarily driven by the accurate profiling of user behavior*. Indeed, any solution with the potential of being realistically effective needs to consider that user data generates the revenue that pays for "free stuff". In this work we present a new perspective on the mobile phone privacy problem by considering the implications of a solution for the current business model that drives the free application market. An analysis of the Android application market reveals that the abundance of free applications is likely to be attributed to the failure of the existing privacy protection mechanisms available on the smart phone platforms. By analyzing the current business model in mobile application advertisement, we show that a wide adoption of more rigorous privacy protection mechanisms can potentially lead to the collapse of the ad-driven mobile application market. We then present our efforts towards a realistic solution to a market-aware privacy protection framework. Our approach is based on the establishment of a feedback control loop that adjusts the level of privacy protection on mobile phones, in response to advertisement generated revenue.

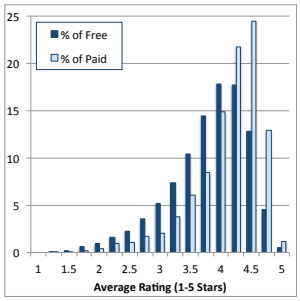
2. MARKET ANALYSIS

The smart phone industry has employed a range of mechanisms to address the risks involved with the installation of third-party applications on mobile phones. The two leading smart phone operating systems follow different approaches. In Apple's iOS, all new applications are required to undergo Apple's closed approval process before being released to the market. In this model Apple acts as a trusted party that gives a *one-off certificate* that the application can use to retrieve information from a mobile phone. After an iOS application is released, no information is offered about the types of data the application has access to (with the exception of location data where user access control is allowed). Android avoids the need for a trusted party by allowing the privacy negotiation to take place directly between the developer and the user. In Android every

HotMobile'12 February 28–29, 2012, San Diego, CA, USA.

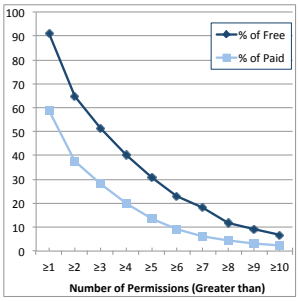


(a) Number of downloads (CCDF)

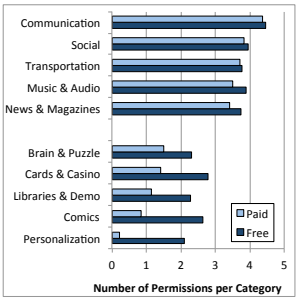


(b) Average Rating (Histogram)

Figure 1: Application popularity.

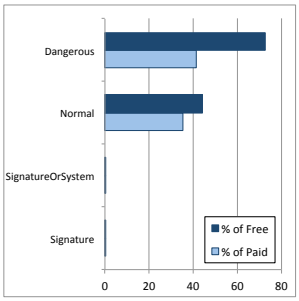


(a) Distribution of permissions requested (CCDF)

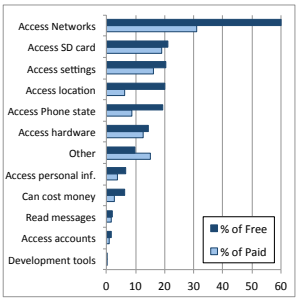


(b) Number of permissions for the top/bottom 5 categories.

Figure 2: Number of permissions requested.



(a) Permission groups



(b) Dangerous permission groups

Figure 3: % of apps requesting at least one permission in a category.

application is required to explicitly specify its permission requirements. During installation the user is presented with a grouped list of permissions that the application requires to run. Such permissions include access to accurate or coarse grained location, the phone's state, access to the phone's address book, to name a few. The assumption is that based on the requested permission list, the users can make an informed decision about whether they want to install the application or not. Effectively, the Android permission model involves a one-off control point (installation) after which the application has full access to the requested resources.

In order to understand the impact that the existing permission model has on the mobile application landscape, we explored the ecosystem of the Android application market. We investigated the picture of the Android Market by analyzing the metadata of all the applications available on the online market. The market crawl was performed through a Java-based tool during July 2011. Using 64 Google accounts and a combination of targeted search queries, we were able to capture the whole market over a six week crawling period. The collected data include the full set of metadata for 251,342 applications[1]. The metadata include, amongst other information, the application title, type, category, number of downloads, average rating, and the full list of requested permissions.

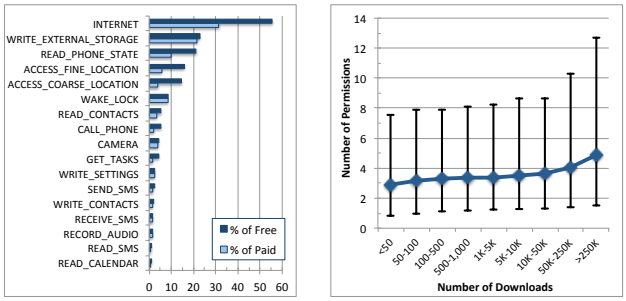
The complete market consists of 73% free applications. In almost all of the application categories the high ratio of free vs paid applications is evident. The exceptions are categories that may include copyrighted content such as the "Personalization" category (26% free) with a large number of wallpaper applications, and the "Books & References" category (53% free). A comparison between free and paid applications shows somewhat expected results: free applications are significantly more popular than paid ones (Figure 1(a)), with 20% of free applications downloaded more than 10,000 times, in contrast to only 0.2% of paid. At the same time paid applications receive higher user ratings (Figure 1(b)).

The high popularity of free applications offers an indication of the possible monetization opportunity through advertisement. In order to explore any possible trend we manually inspected the 50 most popular free applications in the market. Of those 50, 11 applications do not use advertisements, 32 applications use advertisements through add-on widgets and 7 applications show advertisements integrated with the application. Overall, 77% of the top free applications were ad-supported.

Higher advertisement revenues are typically achieved through targeted advertisements. As a consequence, better profiling of the target user requires access to more personal information about the owner of the device. We explore such trends in the Android market by inspecting the access permissions requested by free applications. Figure 2(a) shows the distribution of permissions requested by the Android applications. Surprisingly, 40% of the paid and 10% of the free applications do not require any permission to run. Most of these applications are themes, comics, games, and development libraries (Figure 2(b)). Furthermore, 7% of the free applications request more than 10 permissions compared to only 1.8% of paid ones. Our results indicate that, on average, free applications usually request 2-3 additional permissions compared to paid applications of the same category. It is interesting to note that incorporating targeted advertisement offered by a popular Ad-network such as AdMob would require permissions for Internet access, location, and device identification, leading to 3 additional permissions for an application that does not require them by default.

The Android OS organizes permissions into different threat levels: *normal* permissions are considered of minimal risk and include access to normal functionality, e.g. access to the vibrator, reading the battery level. *Dangerous* permissions are considered of high risk and could pose a threat to the user's privacy. Such permissions include access to location and access to SMS content. When an application requests a dangerous permission, a warning is displayed during installation to alert the user of the potential risks. The user is expected to make a binary decision on whether to install that application or not. Finally, there are signed or system permissions that are not presented at all to the user. The analysis on the market data shows that the majority of applications request at least one dangerous permission (Figure 3(a)). Furthermore, if we consider the difference between paid and free applications, it seems that free applications mostly request additional dangerous permissions: 73% of free applications request at least one dangerous permission compared to just 41% of paid ones. According to this trend, *7 out of 10 free applications would show a warning to the user during installation*. It is well established that such types of frequent warnings render them completely ineffective, as users tend to keep acknowledging them without much consideration [15]. This could imply that applications can request highly sensitive information unchallenged by the end users, thus defeating the purpose of Android's permission model. For instance, the frequency of such warnings might distract the users from noticing that 7% of the applications request access to user contacts (Figure 4(a)).

[1]According to the official Google blog at August 2011 there were 250,000 applications in the Market.



(a) 17 most popular permissions. (b) # of downloads vs # of permissions.

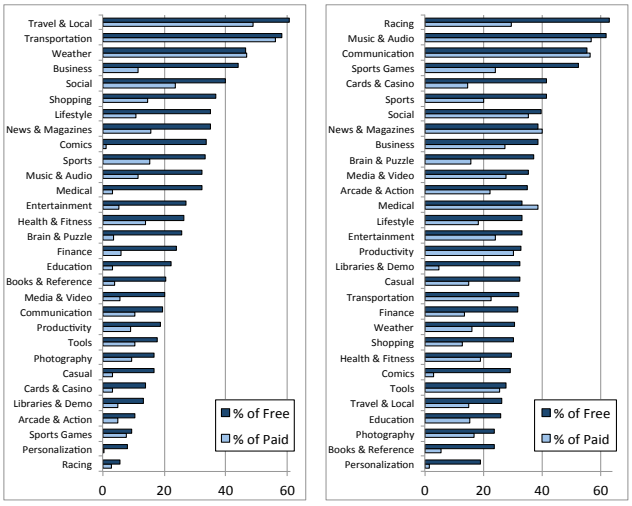
Figure 4: Permissions and popularity.

In Figure 5(b) we show the percentage of applications that request at least one permission that could reveal personal information. This includes applications that ask for access to read messages, accounts, phone calls, the user's contacts, phone state (provider, phone number, IMEI, etc), etc. Surprisingly, games seem to frequently request these permissions (e.g., more than 60% of free racing games), followed by social and multimedia applications. On average, an alarming portion of applications (30%) request at least one of these permissions. Again, we observe that free applications are more likely to request such information. We should note here that more than 94% of these applications also request network access and, therefore, could potentially leak this information.

In Figure 5(a) we consider the statistics for location information. While the top categories are somewhat expected ("travel & local", "transportation", "social" and "weather"), we can also see amongst the top ranking categories "comics", "games" and "finance". Furthermore, paid applications in the same categories do not demonstrate the same trend. For instance, only 1% of the paid *"comics"* require location compared to 33% of the free versions. The same pattern occurs for other applications such as books, personalization, medical education, and for games. We deduce that a high percentage of free applications request location access that is not justified for their particular application category. To examine the extent to which access to location is unrelated to the service offered by free applications we analyzed the descriptions of all applications that request access to location. A text filter was used that matched the presence of 22 key phrases that are associated with location based services (e.g. location, navigate, local, GPS, nearby). Although the filter was not intended to be exhaustive, the comparative results between paid and free applications are revealing. 62% of the paid applications requesting location include one of these keywords in their description, compared to only 32% of the free ones.

Finally, we examined whether the permission model followed by Android is able to put pressure on developers to avoid requesting unnecessary permissions by making the users reluctant to install applications that ask for risky permissions. We explored the market to find any information that could indicate that such a relation exists. A comparison between the popularity of applications and the number of permissions they request showed that there was no correlation between the two. Figure 4(b) shows that there is actually a slightly higher number of permissions on average for the most popular applications, although the high variability does not allow us to make any definite claims.

In summary, our analysis shows that the Android Market is mostly composed of highly popular free applications. As free applications are primarily supported by advertisements, this further indicates that the whole ecosystem depends on this revenue model. Furthermore, our results show that the current permission-based mechanism fails to deter developers as free applications request significantly more permissions and there is no correlation between popularity of an application and the amount of sensitive information requested. This can be attributed to the nature of the existing warning model that may distract the users. Finally, as our results indicate that free applications request significantly more private permissions than paid applications, it is important to examine how the current mobile advertisement model works.



(a) Location. (b) Other private information.

Figure 5: % of apps requesting sensitive information.

3. MOBILE ADVERTISEMENT MODEL

The mobile advertising business model is the primary mechanism that funds the exploding market of mobile phone applications. We investigate this business model, through an experimental analysis of a popular ad-network (AdMob). In its generic form the model involves three main parties: the *user* who is the recipient of a service delivered by a mobile application; the *developer* who expects to be rewarded/compensated for delivering the service; and the *ad-network* that compensates the developer in exchange for the successful gathering of user interest in businesses through targeted adverts. AdMob pays a developer according to the number of impressions a certain advert has on his mobile application, with significantly more funds offered when an impression generates a "click" from the user. The actual price that is offered depends on the demand of different companies to advertise on a particular slot.

In this model the ad-network has a strong incentive to generate as many clicks per impression as possible for its adverts in order to satisfy their clients and to minimize the cost of unsuccessful advertisement impressions that should be paid to developers. This means that the ad-network should target users that are most likely to find them useful. Advertising networks are generally secretive about the profiling algorithms employed for targeted advertisement. However, it is understood that demographic information (gender, age), location, online behavior, and social networks are some of the information used in such algorithms. Moreover, with the advent of ubiquitous computing, the user's whereabouts are extensively used to display location-based advertisements [11]. It is then clear that an advertising network has strong incentives to collect as much information as possible about a user. However, in typical in-application advertisement frameworks, this collection of profile information is delegated to the target application that is then responsible for feeding it to the ad-network (Figure 6(a)). Therefore, the current model relies on the fact that developers have a strong incentive to support the accurate profiling of the user, as this can increase their revenue.

However, the tight coupling of the ad-support widget with the application means that permissions that are required to support targeted advertisement are publicized as application permissions, merely to allow the passing of private information to the ad-network (Figure 6(a)). The end result is a market where the privacy re-

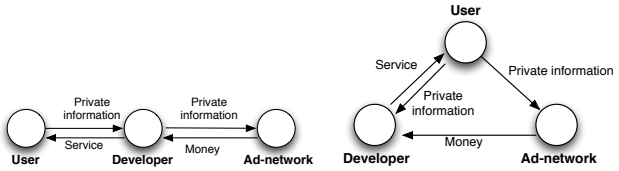
quirements of the application rarely reflect the actual needs of the service that is delivered. This model disrupts the trust expectations of the user by obscuring *who* might access its information, and *how* that information might be used. Our market analysis indicates that this disruption leads to a general disregard of the permission control mechanism by the user. The current permissions model does not seem to impose any significant pressure on developers to limit the number of permissions they request for their applications. In essence the existing model is pushing the market for an uncontrollable collection of as much personal information as possible, without any indication of a force governing this.

4. PRIVACY PROTECTION

Acknowledging the problem of privacy preservation in mobile phones, a number of approaches attempt to limit the flow of private information towards third party applications. MockDroid [2], and Apex [12], for example, are systems advocating the blocking of the flow, with the delivery of false information to certain applications, while [14] considers location privacy with the addition of noise to the delivered information. In the general form, most solutions attempt to limit the flow of private information to the application, without however acknowledging that part of that flow is used for the profiling of the user for targeted advertisement. Indeed, such techniques do not blend well with the ecosystem of the ad-supported mobile application market. The tight coupling of access to private information and ad-supported revenue means that privacy control could have grave consequences for the free application market. Allowing individuals to withhold as much private information as they want, will lead to a starvation of the profiling stream and, consequently, a diminishment of the revenue stream for the developers of ad-supported applications. This would eventually lead to an unsustainable free application market. This particular scenario can be considered a specific case of the "tragedy of the commons" [7] where individuals acting selfishly will deplete a shared resource when it is clear that it is not in anyone's long-term interest for this to happen. Any viable solution to the privacy problem, should aim for a system achieving some sort of equilibrium, where the flow of private information is enough to sustain the flow of service delivered to the users. In the following paragraphs we identify key design strategies that can lead to a sustainable solution for privacy control in an ad-supported mobile application market.

Decoupled Model: Our proposed framework for market-aware privacy control is based on decoupling privacy control between the application and the advertisement support component (i.e., to replace the model shown at Figure 6(a) with the model shown in Figure 6(b)) where two separate flows of information are allowed: one towards the application/developer and one towards the ad-networks. The decoupling allows the specification of *distinct privacy requirements* for the two entities. For the application, this allows the specification of privacy requirements that are directly related to the actual service offered by the application. For the ad-network component, the distinct flow of private information can allow the implementation of privacy control techniques specifically designed to support an ad-driven market. This separation allows users to commit to different sharing agreements between the two entities as the information that is shared with advertisers could be subjected to stricter privacy regulations similar to those that are enforced to telecommunication providers, ISPs, banks, etc. Furthermore, this separation restores the user's trust expectations as it clarifies who is the actual recipient of each different flow of private information. The users can then more actively control the offered information based on how much they trust each entity. This model requires two internal mechanisms to work:

- *Causal link between revenue and privacy:* The decoupling of application and ad-network leads to a clear three-party model, where the flow of private information from the user to the ad-network is responsible for the success rate of targeted advertisement and therefore the revenue flow to the developer, who, as a result, provides the service (the clockwise cycle shown at Figure 6(b)). In order to establish a mechanism that can potentially achieve an equilibrium between revenue and privacy, a channel that creates a causal link between the revenue flow and the private information flow needs to be devised. We discuss possible such mechanisms in Section 4.1.
- *Incentivizing Developers:* Developers might be reluctant to adopt a new advertising model when the current model allows them to greedily collect as much information as possible without any evident impact on the popularity of their applications. Therefore, putting in place mechanisms that prevent developers from collecting unnecessary user data, could incentivize them to adopt the split-flow model proposed. Within the split-flow model such mechanisms can prevent developers from collecting unnecessary user data that are not used by the developed application. Our proposal is that the market should be redesigned in a manner that would enhance awareness and allow users to impose peer-pressure on developers in order to separate their revenue flow and request less permissions for operation purposes. These mechanisms will not only encourage developers to adopt the new model, they will also be used in the newer model to discourage developers from requesting unnecessary permissions (i.e., control the information flow between users and developers in Figure 6(b)). We analyze such solutions in Section 4.2.



(a) The current implementation. (b) The envisaged model.

Figure 6: Mobile Advertisement Models.

4.1 Market-aware Privacy control

The main challenge in building a market-aware privacy control framework is to establish a causal link between the flow of private information that is delivered to an ad-network and the flow of revenue delivered to the developer. The decoupling of the application and ad-network components allows us to individually manipulate the flow of private information to the ad-network, without affecting the service offered by the application. Controlling the private information flow requires the presence of a privacy *firewall* between the source of private information (the mobile OS) and the ad-network component. Such a privacy firewall, provides a gradient of privacy flow control states. We assume that the level of private information that is given to an ad-service can depend on a range of variables, such as information accuracy or timelines. In the case of location, for example, different levels of privacy control can lead to more or less frequent updates, with more or less accurate location information. According to the analysis of the advertising market, the level of detail of the private information that is given to an ad-network is positively correlated to the possibility of a target advertisement to generate a successful "click". Consequently, reducing the level of detail given to the ad-network is deemed to cause a knock-on effect on the number of successful clicks.

In order to avoid the starvation effect that can be caused by an uncontrolled reduction of the flow of private information, it is necessary to forbid users from directly controlling it. Instead, the flow of private information should be linked with the resulting revenue. The actual revenue that a developer generates through advertisement is not publicly accessible. However, a measurement of the number of advertisement "clicks" generated by an application is a strong indicator of the revenue flow.

The proposed solution consists of a framework that allows the aggregation and sharing of the number of generated “clicks" on each ad-supported application. The shared information about the level of success of targeted advertisement is then used to change the flow of private information delivered to the ad-network. This creates two opposing forces: if the click-through ratio (and therefore the revenue) of a given application is high, the users have the opportunity to limit the private information leaked to the ad-network. If the private information flow is restricted too much, the drop on revenue flow triggers an increase on the private information flow.

The design of the system consists of a dynamic monitoring component operating on the mobile phone that measures the “click-through" ratio for a given application. The collected measurements are aggregated through a market-server. Similarly, the privacy control component synchronizes with the market-server, in order to adjust the level of privacy flow for all users of the given application. As the privacy control mechanism operates over the aggregate click-through ratio, no private information about individuals needs to be maintained by the market server. More details on how this can be implemented can be found in Section 5.

The revenue threshold problem The proposed framework creates a feedback control mechanism that maintains the flow of private information for a given ad-supported application. For an application the revenue flow is dependent of the number of users that are running an application and the average click-through ratio triggered by advertisements. One of the challenges in this mechanism is the establishment of a desired “click through" ratio considered acceptable for an application. That ratio will govern the adaptive changes to privacy flow control, increasing the flow when the click ratio is low and vice versa. Different market strategies can be implemented, ranging from a fixed click through ratio for all applications (allowing popular applications to achieve higher revenues by the higher number of users), or by constructing appropriate incentive driven mechanics and allowing the developers to specify a “value" for their application in the form of ad-supported revenue. In the latter case, the ad-supported value can be set in a similar manner as the price that developers set for paid applications. In such case users can decide whether they would like to offer the requested information (e.g., location, device ID) to support the requested ad-supported revenue, to purchase the paid version or anything in-between. Furthermore, incentive mechanisms such as those described in [8] can be used to enhance user participation. The introduction of such mechanics in the ad-supported applications would lead to new dynamics on the mobile application market.

4.2 Incentivising Developers

The introduction of a new advertisement framework bears the major challenge of incentivizing current developers to adopt the new model. As the market analysis reveals, the existing privacy control mechanisms offer a fertile environment for an abusive ad-supported market, with uncontrollable collection of private information. The proposed framework for market-aware privacy control requires developers to adopt a novel advertising approach that could potentially limit their revenue flow in order to balance access to private information. Clearly, the proposed model is not likely to be adopted unless some additional mechanisms are employed to encourage developers to switch.

We therefore propose an approach which aids to limit the developers from setting arbitrary permissions by establishing a mechanism to apply peer pressure on the developer community with respect to the privacy requirements of their applications. Such mechanisms would apply significantly more pressure to the current in-application advertisement model (Figure 6(a)) which expects developers to request more privileges than required by their applications in order to support targeted advertisements. Our expectation is that this would eventually pave the way for the adoption of a split-model (Figure 6(b)), where their applications are judged independently of the ad-support mechanics and the users have clearer expectations about how their information is used. Furthermore, within the newly proposed model such mechanisms are still crucial for incentivizing developers against harvesting private user information for their own use (the flow between users and developers in Figure 6(b)).

Our solution relies on two components:

- A privacy monitoring component that can track the private information collected by any application operating in a mobile phone (e.g., TaintDroid [3]). The purpose of the privacy awareness component is to allow user to understand what information is captured by any given application running on their phone.
- A privacy aware application market where users can vote on the necessity of individual privacy requirements for certain applications. This will allow existing users to effectively flag suspicious permissions that, according to their own criteria, are not necessary for operational purposes. The crowdsourced information on privacy requirements can be integrated with the installation process of applications in order to produce meaningful warnings about over-privilledged applications, possibly impacting the popularity of applications with unreasonable privacy requirements.

5. IMPLEMENTATION

We explored the feasibility of our ad-aware privacy control framework by modifying the Android operating system. Our implementation is based on Cyanogen v2.3.4: a community supported variant of Android OS that can be installed to a large variety of devices.

Separating Permissions for Advertisements:

The decoupling of application and advertising permissions was achieved by separating the two functions into distinct binaries. We implemented a generic advertising service that is installed separately, requesting its own set of permissions such as location and Internet access. The advertising service exports a new `Intent`[2] in order to allow other applications to subscribe. Applications that require the use of the advertising service need to request a *user-defined* permission: `ACCESS_ADVERTISEMENT_SERVICE`. Communication between the application and the advertising service is performed through a client-side library that is compiled with the target application. Through Android’s IPC, the application can launch a widget that illustrates advertisements on top of their UI, without the need to specify any permissions that are required for targeted advertisement.

Real-Time Monitoring: The real-time monitoring is part of the mechanism that controls the flow of private information between the mobile device and both the running application and the advertising service. The mechanism is implemented through a system service that is part of the OS and is capable of intercepting all interactions between two applications and between an application and system services. To do this we modified various OS components (e.g., the `Binder` and the `ActivityManager` services). This service records access patterns to critical components such as attempts to get the user’s location, SMS, phone number or device ID. Furthermore, for the advertising service, the real-time monitoring captures the number of times a user may “click" on an advertisement, as well as the amount of time an advertisement is displayed. This information is collected as part of the privacy-control loop mechanism and is used to i) dynamically control how much information is exposed and ii) increase user awareness.

Dynamically control exposed data: To control access to private information we implemented a new system service. This service can dynamically revoke a permission to an application, or intercept and anonymize the return value of an ICC call, intent or content

[2]Android’s event-based interprocess communication mechanism.

provider. The service can independently block or anonymize information depending on whether the information is requested by an application or an advertising service as discussed in Section 4.

To obfuscate the user's location information we modified Android's `LocationService`. For each application the user can reveal the real location, a fuzzy location (e.g., a random location within a given range from the real location) or return *unknown location*. The amount of added noise can be automatically calculated in order to maintain the revenue model. Moreover, as it has been shown that people value location information differently based on where they are [9], in our prototype the user can create *location-based rules* to obfuscate location when in certain locations (e.g., hide the exact location when at home). Similarly, our prototype can anonymize the user's identity (IMEI, Phone Number, Contacts, SMS etc) while protecting the developer's revenue by providing a consistent random response to an application.

Increasing awareness and exerting peer pressure: We are currently developing a website that will replicate the functionality of the Android Market. We use the information that we crawled for more than 250,000 applications. For a given application, the users will have the opportunity to vote and comment on each of the requested permissions. Furthermore, the developer will have the opportunity to write a short description (less than 160 characters) about why this permission is required. Furthermore, we are developing an enhanced version of the installation manager that will work in conjunction with the enhanced market. During installation the user is be presented with a sorted list of permissions. The sorting depends on the voting: permissions that other users flagged as unreasonable will be displayed first. Finally, to display the real-time data that we gather we implemented a simple graphical user interface where a user is shown a list of all installed applications and information about access patterns to sensitive data such as the location, contacts, messages, device ID, IMEI number.

We soon plan to launch the real-time monitoring tool, the website and the installer as we would like to invite users and developers to participate in a large-scale user study that aims to better understand the current permission model and to validate the effectiveness of such solutions. However, we believe that such a model should preferably become an integrated part of the existing ecosystem.

6. RELATED WORK

As mentioned, a number of papers have studied the current status of mobile phone applications permission access and many approaches have been devised to limit the access to users' data. In [1] the authors analyzed 1,100 Android applications in order to group their permission patterns. Stowaway [4] analyzed the byte-code of 940 applications and found that about one-third of them are over-privileged. Similar results were found in [5] and [16], where the authors designed a tool that aids the developers in specifying a minimum set of permissions. While we focused on the differences between free and paid applications by examining more than 250 thousand applications, these works further support our findings that application developers are not deterred by the current permission model and they often request more permissions than necessary.

TaintDroid [3] is a framework able to track how sensitive data could leak to the Internet. The authors used this system to monitor the behavior of 30 popular Android applications and they showed that 20 of these applications might misuse users' private information. TaintDroid also revealed that 15 of the 30 applications reported the users' locations to remote advertising servers while 7 applications collected the device ID, phone number and the SIM card's serial number and sent them to the developer. These results reinforce our findings and provide another reason why a privacy protecting framework is necessary.

Finally, Apex [12] is a framework that extends the Android permission model by allowing users to selectively revoke a permission or impose constraints on the usage of resources (e.g., number of SMS) at install time. Similarly, MockDroid [2] allows users to provide fake data to an application. While these approaches are very powerful in limiting user data leaks, they might damage the current revenue model for free applications. Our approach is an alternative which tries to keep this revenue model in mind while controlling user information flow to developers and ad-network.

7. CONCLUSIONS

In this work we are presenting a new perspective on the problem of privacy protection for mobile phones. By analyzing more than 250,000 applications of the Android market we identify the significance of targeted advertisement in the success of the free application market for smart phones. Identifying the tight link between access to private information about users and ad-supported applications, we illustrate that privacy protection solutions bear the risk of breaking the current business model that offers financial support to the developers. Our prototype solution aims to introduce a privacy control mechanism that is interlinked with the ad-supported revenue delivered to free applications. By establishing a feedback control loop, privacy control can be dynamically adjusted in order to maintain an equilibrium between the flow of private information and the generated advertisement revenue. Our intention is to release our model as an alternative advertisement service and evaluate the impact of this mechanism in the mobile application market.

Acknowledgments: We would like to acknowledge the support of the EPSRC through project FRESNEL EP/G069557/1. We also thank Alastair Beresford, Chloe Brown, Paolo Costa, Daniele Quercia, Salvatore Scellato, Franck Stajano and the members of the NetOS group for their feedback.

8. REFERENCES

[1] D. Barrera, H. G. u. c. Kayacik, P. Oorschot, and A. Somayaji. A methodology for empirical analysis of permission-based security models and its application to android. In *CCS '10*, NY, USA, 2010. ACM.

[2] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan. MockDroid: trading privacy for application functionality on smartphones. In *HotMobile 2011*, Mar. 2011.

[3] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. In *OSDI'10*, Berkeley, CA, USA, 2010.

[4] A. P. Felt, E. Chin, D. S. Steve Hanna, and D. Wagner. Android permissions demystified. In *University of California, Berkeley, Technical Report No. UCB/EECS-2011-48*, May 2011.

[5] A. P. Felt, K. Greenwood, and D. Wagner. The effectiveness of application permissions. In *WebApps'11*, Berkeley, CA, USA, 2011.

[6] S. Guha, B. Cheng, and P. Francis. Privad: Practical Privacy in Online Advertising. In *NSDI'11*, Mar. 2011.

[7] G. Hardin. The Tragedy of the Commons. *Science*, 162:1243–1248, Dec. 1968.

[8] J.-S. Lee and B. Hoh. Sell your experiences: a market mechanism based incentive for participatory sensing. In *PerCom'10*. IEEE, Mar. 2010.

[9] J. Lindqvist, J. Cranshaw, J. Wiese, J. Hong, and J. Zimmerman. I'm the mayor of my house: examining why people use foursquare - a social-driven location sharing application. In *CHI '11*, NY, USA, 2011.

[10] Lookout. Mobile threat report @ONLINE. `https://www.mylookout.com/mobile-threat-report`, Aug '11.

[11] J. Müller, F. Alt, and D. Michelis. *Pervasive Advertising*. Book, Springer Verlag'11.

[12] M. Nauman, S. Khan, and X. Zhang. Apex: extending android permission model and enforcement with user-defined runtime constraints. In *ASIACCS '10*, pages 328–332, NY, USA, 2010. ACM.

[13] P. Nickinson. Android market now has more than a quarter-million applications. Android Central: `http://www.androidcentral.com/android-market-now-has-more-quarter-million-applications`, July 14 2011.

[14] D. Quercia, I. Leontiadis, L. McNamara, C. Mascolo, and J. Crowcroft. Spotme if you can: Randomized responses for location obfuscation on mobile phones. In *ICDCS'11*, Los Alamitos, CA, USA, 2011.

[15] D. W. Stewart and I. M. Martin. Intended and unintended consequences of warning messages: A review and synthesis of empirical research. *Journal of Public Policy Marketing*, 13(1):1–19, 1994.

[16] T. Vidas, N. Christin, and L. Cranor. Curbing Android permission creep. In *W2SP 2011*, Oakland, CA, May 2011.