Exposures Exposed: A Measurement and User Study to Assess Mobile Data Privacy in Context

Evita Bakopoulou  
University of California, Irvine  

Anastasia Shuba†  
University of California, Irvine  

Athina Markopoulou  
University of California, Irvine

Abstract

Mobile devices have access to personal, potentially sensitive data, and there is a large number of mobile applications and third-party libraries that transmit this information over the network to remote servers (including app developer servers and third party servers). In this paper, we are interested in better understanding of not just the extent of personally identifiable information (PII) exposure, but also its context (i.e., functionality of the app, destination server, encryption used, etc.) and the risk perceived by mobile users today. To that end we take two steps. First, we perform a measurement study: we collect a new dataset via manual and automatic testing and capture the exposure of 16 PII types from 400 most popular Android apps. We analyze these exposures and provide insights into the extent and patterns of mobile apps sharing PII, which can be later used for prediction and prevention. Second, we perform a user study with 220 participants on Amazon Mechanical Turk: we summarize the results of the measurement study in categories, present them in a realistic context, and assess users’ understanding, concern, and willingness to take action. To the best of our knowledge, our user study is the first to collect and analyze user input in such fine granularity and on actual (not just potential or permitted) privacy exposures on mobile devices. Although many users did not initially understand the full implications of their PII being exposed, after being better informed through the study, they became appreciative and interested in better privacy practices.

1 Introduction

Mobile devices have access to a wealth of personal, potentially sensitive information and there is a growing number of applications that access, process and transmit some of this information over the network. Sometimes this is necessary for the intended operation of the applications (e.g., location is needed by Google Maps) and controllable (e.g., by the user through permissions), but for the most part, users are not in control of their data today. Applications and third party libraries routinely transmit user data to remote servers, including application servers but also ad servers and trackers, and users have typically limited visibility and understanding of what part of their personal data is shared, with whom, and for what purpose.

With the increased interest in online privacy, there are several bodies of related work. On one hand, a number of systems have been proposed that improve data transparency and protect personally identifiable information (PII). In general, these systems fall into three categories: (i) static analysis and application re-writing [15, 18, 26], (ii) dynamic analysis with a modified or rooted OS [7, 16, 17, 41, 46], and (iii) VPN-based network monitoring [27, 29, 32, 35–38]. While these tools provide more fine-grained control over sensitive data (as opposed to just permissions), the way users engage with these privacy-preserving systems is less well studied. On the other hand, in the human-computer interaction (HCI) community, researchers have extensively studied how different designs for app permissions affect users’ decisions on which apps to install and which permission requests are considered legitimate by users [10, 19, 20, 22, 44]. A detailed review of related work is provided in Section 2.

In this paper, we are interested in understanding privacy exposures, which we define as PII transmitted by a mobile app (or third party library used by the app) on the device, over the network interface, towards a remote server. Our goal is to understand not only the extent and mechanisms of PII exposure, but also its context (i.e., functionality of the app, destination server, encryption used, frequency, etc.) and the risk perceived by mobile users today. For example, location needs to be shared for a navigation app to perform its intended and legitimate function, and should not be of concern to the user. In contrast, if the same navigation app uses a library that shares device ids with a third-party server, this is
more likely a privacy leak\textsuperscript{1} and should be of concern to the user. We are also interested in PII actually exposed in real network traffic, as opposed to potential privacy exposures as captured by permissions. To that end, we make the following two contributions.

**Measurement Study.** First, we utilize a state-of-the-art mobile network monitoring tool, AntMonitor [35], to intercept and inspect outgoing packets transmitted over the mobile device’s network interface. Using AntMonitor, we conduct two extensive and systematic experiments (one manual and one automated) on Android phones, where we test 400 most popular Android apps (as of March 2017) and we collect 47,076 outgoing packets. We identify whether these packets contain any of 16 predefined types of PII (defined in Section 3.1), together with related information, which we collectively refer to as context including: the destination server/domain (i.e., whether it is an App Developer server or a third-party Advertisers & Analytics server), the app category (games, shopping, navigation etc.) which reveals the intended functionality of the app, whether the PII is exposed in clear text or is encrypted, and whether the app runs in the background or foreground. Our datasets partly confirm findings of previous measurement studies of mobile devices but are richer: e.g., they contain previously unseen exposures over plain TCP and UDP, exposures while the app is in the background, and malicious scanning for rooted devices. We analyze our datasets and provide insights into the extent and nature of how PII is exposed today. We also identify behavioral patterns, such as communities of domains and mobile apps involved in exposing private information. These patterns can be used in the future to design automated prediction and prevention methods. We plan to make the datasets available to the community.

**User Study.** Second, we perform a user study on Amazon Mechanical Turk (MTurk) with 220 users. We summarize the results of the measurement study in categories, present participants with real-world scenarios of private information exposure in context (type of PII, whether it is shared with the application or a third party, use of encryption, etc.) and we ask them to assess the legitimacy (i.e., whether the information is needed for the app’s functionality) and privacy risk posed. We also educate the participants on how a single piece of PII can lead to even more information being discovered when combined with data fetched from a data broker. Finally, we ask users before and after the survey what actions they would be willing to take to protect their privacy, including using free/paid privacy-enhancing tools and contributing their data to crowdsourcing. To the best of our knowledge, our user study is the first to collect and analyze user input in such fine granularity (i.e., taking context into account) and on actual (not just potential or permitted) privacy exposures from mobile devices. We found that (i) many users did not initially understand the full implications of their PII being exposed but (ii) after being better informed through the study, they became appreciative and interested in better privacy practices. The insights gained by the study can inform the design of fine-grained data transparency and privacy preserving tools such as AntMonitor [2].

The structure of the rest of the paper is as follows. Section 2 reviews related work. Section 3 describes the measurement study, including the data collection, summary and analysis. Section 4 presents the Amazon Mechanical Turk study, based on our datasets. Section 5 concludes the paper and outlines directions for future solutions based on the findings of this study.

2 Related Work

**Privacy-Preserving Tools.** There are a number of complementary frameworks, built by different communities, that can enhance data transparency and privacy protection on mobile devices.

Permissions are the first line of defense against unwanted access to sensitive resources. However, they are insufficient and too coarse-grained: (i) users typically accept to install apps by default; (ii) permissions do not capture run-time behavior; (iii) they do not protect against inter-app communication and poorly documented system calls; and (iv) permissions signify access to information, which is less of a concern than sharing that information over the network.

Static analysis and application re-writing, such as PiOS [15], AndroidLeaks [18], and [26], suffer from the inherent imprecision of decompilation. Furthermore, static analysis does not capture representative run-time behavior and often fails to deal with native or dynamically loaded code.

Dynamic analysis with a modified or rooted OS include ProtectMyPrivacy [13], TaintDroid [16], and others [7, 10, 17, 41, 43, 46]. Such tools are powerful but not suitable for mass adoption since rooting a phone or installing a custom OS is not only a daunting task for the average user, but is also strongly discouraged by wireless providers and phone manufacturers.

VPN-based network monitoring tools use the TUN interface to capture network packets on the device and detect whenever sensitive information is sent over the network. Over the years, their implementation has evolved from a client-server implementation [27, 32, 36] to a mobile-only implementation [29, 35, 38]. State-of-the-art mobile-only network monitoring tools include AntMonitor [35], Recon [32], Lumen [30], and Privacy Guard [39]) and are amenable to

\textsuperscript{1}Most prior work [31, 32, 36] refers to PII found in outgoing packets as a “privacy leak,” because PII is by definition private information and an outgoing packets indicates exfiltration or a “leak.” However, we purposely distinguish between “privacy exposure” (a PII contained in an outgoing packet) and “privacy leak” (which is an exposure that is not necessary for the intended functionality of the app, and/or goes to a third party server, or happens in clear text). This distinction, between a PII exposure and an actual leak, can only be made based on the context, which is one of the main aspects we investigate in this paper.
crowdsourcing thanks to their implementation as a mobile-only user-space app. However, to the best of our knowledge, they have only been used so far to collect packet traces and analyze “privacy leaks” found therein, not for user studies.

**Privacy “Leak” Datasets.** The aforementioned tools have been used before to collect datasets containing privacy “leaks” (see Footnote 1 for terminology) from mobile devices. Most closely related to this paper, the VPN-based monitoring tools only used so far to collect packet traces and analyze “privacy leaks” found therein, not for user studies. Recon [32] collected packet traces to train machine learning classifiers for predicting PII exposures. Razaghpahah et al. [31] collected cases of PII exposures from thousands of users using the Lumen app and used that data to find new advertising and tracking services. Finally, the longitudinal study of PII exposures in [33] demonstrates how privacy evolves across different app versions in Android.

**User Studies.** Several experimental studies have analyzed mobile app behavior and several users studies have analyzed user interactions and mobile usage (e.g., Mehrotra et al. [25], Tian et al. [40], EarlyBird [45] and Xu et al. [47]). Most closely related to this paper, are user studies that focus specifically on privacy.

Permissions and how users interact with them have been extensively studied in [10, 19, 20, 22, 23, 44]. Almuhiemidi et al. [10] studied how sending users privacy nudges affected their permission settings. Wang et al. [44] studied user decisions when presented with permission settings that are separated between apps and ad libraries. More recently, Ismail et al. [19] showed that it is possible to maintain app usability even when disallowing certain permissions. Chitkara et al. proposed a retrofitted Android system, ProtectMyPrivacy [13], that allows users to make fewer privacy decisions by setting permissions based on third-party libraries instead of applications.

Taking a different approach, PrivacyStreams [21] proposes and evaluates (with a user study) a tool for developers to write code in a more privacy-preserving way. In the web ecosystem, the work in [28] studies online privacy in websites to identify mismatched user expectations and the factors that impact these mismatches. The work by Kleek et al. [42] is closest to our work in that they use information captured from network monitoring to see if it influences users’ decisions to install apps. Unlike their work, however, we are interested in learning what users would do if given more fine-grained control over their data.

**Our work in perspective.** In this paper, we use one of the aforementioned state-of-the-art VPN-based monitoring tools, AntMonitor [2, 35], to collect and analyze packet traces and the privacy exposures found therein. This approach has the advantage that it captures actual real-world privacy exposures, as opposed to e.g., potential exposures described by permissions. In addition, we compile the large volume of information from the packet traces and present it to users in a way that they can process and assess. The combination of a measurement study (volume and coverage of packet traces obtained through extensive and systematic experiments) with a user study (summarizing the information into categories, defining context, and obtaining fine-granularity feedback from users) is one of the contributions of this paper, in addition to the detailed findings of both studies as outlined in the introduction.

### 3 Measurement Study: Data Collection & Analysis

#### 3.1 Data Collection System: AntShield

We are interested in collecting real-world cases of PII exposure, as captured in packet traces of outgoing packets (as opposed to potential exposures e.g., indicated by permissions). We focus on PII that have been previously defined in related work (e.g., [29, 32]). Specifically, we are interested in detecting PII that belong to the following list:

- Device Identifiers: IMEI, IMSI, Android ID, phone number, serial number, ICCID, MAC Address, available through Android APIs.

- User identifiers: usernames and passwords used to login to various apps (unavailable through Android APIs); Advertiser ID, email (available through Android).

- User demographic: first and last name, gender, zipcode, city, etc. - unavailable through Android APIs.

- Location: latitude and longitude coordinates, available through Android APIs.

To collect real cases of when the PII defined above are exposed, we build on AntMonitor, an app that intercepts all network traffic from the device without requiring rooting. We pick AntMonitor since it is a representative VPN-based tool for privacy protection (see Sec. 2) and it is easy to extend it for data collection. The traffic interception, along with several utility functions of AntMonitor have been made available as a library, and we will refer to it as the AntMonitor Library [2, 35]. We will refer to our data collection app that extends the AntMonitor Library, as AntShield [2, 34, 37].

As shown in Fig. 1(a), AntShield receives outgoing packets via the PacketFilter interface provided by the AntMonitor Library. Note that the AntMonitor Library also implements a TLS proxy, which allows it to decrypt SSL/TLS traffic of applications that do not use certificate pinning (see [35] for details). These decrypted packets are also passed to AntShield via the PacketFilter interface. Each intercepted

---

2The design and performance evaluation of the AntShield system is a contribution on its own. However, we consider it out of the scope of this paper. Here, we only use AntShield as a tool to collect the packet traces that are the starting point of our measurement and user studies.
AntMonitor Library’s mapPacket API call to note which app was responsible for generating the outgoing packet in question. Finally, we break the packet into any relevant fields (destination IP address/port, HTTP method, if applicable, and etc.) and save it in JSON format. Any PII found is redacted before saving the packet to allow us to share the data with the community. Any contextual information (the PII found, along with the application name, and whether or not this application was in the foreground when it generated the packet) is saved in separate JSON fields. Note that although the entire packet is not necessary for the purposes of this paper, the data could be useful later, for instance to train classifiers that predict PII exposures as in [32] and [37].

In summary, using AntShield to capture packets on the device has several advantages compared to previous datasets collected in the middle of the network: (1) we are able to accurately map each packet to the app that generated it; (2) we can keep track of foreground vs. background apps, to see what kind of data apps send while in the background; (3) we gain insight into TLS, UDP, and regular TCP traffic, in addition to HTTP and HTTPS; (4) scrubbing PII and labeling packets with the type of PII they contain is fully automated. Due to these advantages, we were able to collect comprehensive and realistic cases of PII exposures, as described in the next section.

3.2 Datasets Collected

Using AntShield’s packet capturing ability, we interacted with and collected packet traces from 400 most popular free Android apps, based on rankings in AppAnnie [3]. We used a Nexus 6 device for our data collection, and collected two different datasets, depending on how we interacted with apps, as described next. 3

Manual Testing. First, in order to capture PII exposures during typical user behavior, we tested the top 100 apps in batches: we installed 5 apps on the test device and then used AntShield to intercept and log packets while interacting with each app for 5min. After all apps in the batch were tested, we switched off the screen and waited 5min to catch any packets sent in the background. Next, we uninstalled each app and finally, turned off AntShield.

Automatic Testing. We also used the UI/Application Exerciser Monkey [9] to automatically interact with apps. This does not capture typical user behavior but enables extensive and stress testing of more apps. We installed 4 batches of 100 applications each, and had Monkey perform 1,000 random actions in each tested app while AntShield logged the generated traffic. At the end of each batch, we switched off the screen of the test device and waited for 10min to catch additional exposures sent in the background.

These datasets are publicly available at [2].
### Table 1: Summary of Manual and Auto AntShield datasets collected on the device.

|                  | Auto | Manual |
|------------------|------|--------|
| # of Apps        | 414  | 149    |
| # of packets     | 21887| 25189  |
| # of destination domains | 597  | 379    |
| # of exposures detected | 4760 | 3819   |
| # of exposures in encrypted traffic | 1513 | 1526   |
| # of background exposures | 2289 | 639    |
| # of HTTPS packets | 13694| 13648  |
| # of HTTP packets | 6830 | 8103   |
| # of TCP packets  | 867  | 2264   |
| # of UDP packets  | 496  | 1174   |
| # of exposures in UDP | 17   | 12     |

**Summary.** Since the two (Automatic and Manual) AntShield Datasets capture different behaviors, we describe and analyze them separately. The AntShield datasets are summarized in Table 1. Other state-of-the-art datasets include Recon [32] and [33]. Our datasets confirm and extend the findings of previous work, as outlined in the next section. Thus, in addition to being used to generate survey questions in our user study (Sec. 4), our datasets provide valuable insights on their own (Sec. 3.3), and we will make them available to the community.

### 3.3 PII Exposures Found in the Datasets

Our datasets provide us with insights into the current state of privacy exposures in the Android ecosystem. Some of the captured patterns were previously unknown, and are revealed for the first time here. For example, we were able to detect exposures happening in the background, exposures in plain TCP and UDP (not belonging to HTTP(S) flows), and malicious scanning for rooted devices.

**Background Exposures.** AntShield is in a unique position to capture exposures that happen in the background vs. foreground, and other contextual information that is only available on the device. Table 1 shows that there is a substantial number of background exposures (e.g., half of all exposures in the automatic dataset) that should be brought to users’ attention. Digging deeper, we found several interesting patterns in apps that expose PII both in the background and in the foreground. Fig. 2(a) shows how Flow Free, a puzzle game behaves differently in the background vs. the foreground: in the foreground several device identifiers are sent to ad and analytics servers, and in the background, one of the ad servers (mopub.com) also collects the user’s city. Perhaps this information is needed to serve personalized ads based on the user’s location, but it is unclear why it is needed when the application is in the background and no ads are being shown. Another example is MeetMe, an app for meeting people on-line, whose behavior is shown in Fig. 2(b). In this case, the app collects less PII in the background, but is contacting more ad servers. Such findings are concerning, since apps are causing users data usage and are posing privacy risks even when the user is not interacting with the app.

**Non-HTTP Exposures.** Prior state-of-the-art datasets [32, 33] reported only HTTP(S) exposures. Table 1 reports, for the first time, exposures in non-HTTP(S), including plain TCP or UDP packets. Our dataset contains 29 UDP exposures, all of which were exposing Advertiser ID and Location. As shown in Table 2, we also found some apps (mostly games and photo-editing apps) that exposed the Android ID over non-standard
We also show other types of exposures that the particular do-
rection for analyzing the
ntShield dataset. We define two domains to be similar if they
contacted by the same set of applications (see the box on the
right inside Fig. 3). For example, domains A and B are
similar because they are contacted by two apps (app1, app2).
We depict the similarity of domains A and B as an edge on
the graph of domains, at the bottom of the box. This data can
be readily extracted from our trace, together with the type of
information that was transmitted from apps to domains.

The graph depicted on the left side of Fig. 3 shows a pro-
jection of the underlying bipartite graph (middle step in the
box) on domains (last step in the box); the graph is plotted
and analyzed using Gephi [12]. Nodes in this graph represent
domains; the edges indicate similar nodes as per above defini-
tion; the width of the edge indicates the number of common
applications; and the domain color corresponds to the type of
exposed PII. The clusters of domains in the graph are the out-
put of a community detection algorithm, which is a heuristic
that tries to optimize modularity.3

Interesting patterns are revealed in Fig. 3. First, advertising
is the result of coordinated behavior. For example, it is easy
to identify ad exchanges: mopub.com is in the center of all
communication; and inner-active.mobi and nexage.com
are also clearly shown as hubs. All three large communities
on the bottom and left of the graph correspond to ad networks.
Second, on the top left, there is a community of domains that
belong mostly to Google and Facebook, and two domains

(80, 443, 53) TCP ports, such as 8080 or 10086 (a port known
to be used by trojans, Syphilis and other threats [8]). The
destination IPs could not be resolved by DNS, indicating that
the application may have hard-coded those IPs.

**HTTPS Exposures.** Since the usage of encryption is in-
creasing, we also collected and analyzed PII sent over HTTPS.
Table 3 summarizes the exposures we discovered in HTTPS
traffic. The top app com.as.android.article.master is a
news app, thus it makes sense for it to query the user’s city,
perhaps to fetch localized news. However, it is unclear why
the app needs the user’s IMEI (when it already has the Adver-
tiser ID) and the specific longitude and latitude coordinates
of the user. In this case, the city is needed by the app, but
the IMEI and location coordinates are potentially privacy ex-
posures. Another example is com.cmcm.live - it exposes
5 different device identifiers for no apparent reason. Hence,
although well-behaving apps should use HTTPS, they should
also be inspected for potential privacy exposures as not all
information that they gather is necessary for their functional-
ity. We also found that the majority of top domains receiving
PII over HTTPS were ad-related. Although it is expected for
ad domains to receive the Advertiser ID, other PII should not
be collected. These findings motivated us to conduct the user
study in Sec. 4 to crowdsource answers to the question of
when a privacy exposure becomes a privacy leak.

**Checking for Rooted Devices.** We noticed a suspicious
flag called “jailbroken” or “device.jailbroken” exposed by sev-
eral apps (e.g., com.bitstrips.emoji, com.yelp.android, com.zeo-
pcolab.ctr.ads, etc). This flag was found in the URI content
or in the body of a POST method in the packets, and it was
set to 1 if the device was rooted, or to 0 otherwise. In Table 4,
we show the applications that contain this field in our dataset
and the domain to which the “jailbroken” flag is being sent.
We also show other types of exposures that the particular do-
mains and analyzed using Gephi [12]. Nodes in this graph represent
domains; the edges indicate similar nodes as per above defini-
tion; the width of the edge indicates the number of common
applications; and the domain color corresponds to the type of
exposed PII. The clusters of domains in the graph are the out-
put of a community detection algorithm, which is a heuristic
that tries to optimize modularity.

Interesting patterns are revealed in Fig. 3. First, advertising
is the result of coordinated behavior. For example, it is easy
to identify ad exchanges: mopub.com is in the center of all
communication; and inner-active.mobi and nexage.com
are also clearly shown as hubs. All three large communities
on the bottom and left of the graph correspond to ad networks.
Second, on the top left, there is a community of domains that
belong mostly to Google and Facebook, and two domains

### Table 2: TCP packets (non HTTP/S) sending PII over ports other than 80, 443, 53

| App Name                          | Leak Types      | Port  |
|-----------------------------------|-----------------|-------|
| System                            | IMEI, IMSI, AndroidId | 8080  |
| com.jb.gsoms                      | AndroidId       | 10086 |
| com.jiubang.go.music              | AndroidId       | 10086 |
| air.com.hypah.io.slihther          | Username        | 10086 |
| com.jb.emoji.gokeyboard           | AndroidId       | 10086 |
| com.gau.go.launcherex             | AndroidId       | 10086 |
| com.steam.photoeditor             | AndroidId       | 10086 |
| com.jb.zcamera                    | AndroidId       | 10086 |
| com.flashlight.brightestflashlightpro | AndroidId       | 10086 |

| Domain Name                      | Leak Types      | Port  |
|----------------------------------|-----------------|-------|
| 206.191.155.105                  | Username        | 454   |
| 206.191.154.41                   | Username        | 454   |
| 23.236.120.208                   | AndroidId       | 10086 |
| 3g.cn                            | IMEI, IMSI, AndroidId | 8080  |
| 23.236.120.220                   | AndroidId       | 10086 |
Table 3: Summary of applications and domain names with HTTPS exposures in our dataset (manual and auto)

| App Name                        | PII Types                              | # Exposures |
|--------------------------------|----------------------------------------|-------------|
| com.ss.android.article.master   | City, Adid, Location, AndroidId, IMEI | 752         |
| com.cleanmaster.security        | Adid, AndroidId                        | 174         |
| com.paypal.android.p2pmobile    | City, FirstName, LastName, Zipcode, Adid, SerialNumber, AndroidId, Password, Email | 131         |
| com.offerup                     | Adid, Username, FirstName, Location, AndroidId | 114         |
| com.cmcm.live                   | Adid, AndroidId, Location, IMEI, SerializableNumber, IMSI | 114         |
| me.lyft.android                 | City, FirstName, LastName, SerializableNumber, Zipcode, PhoneNumber, Location, AndroidId | 112         |
| com.pinterest                   | Adid, AndroidId                        | 111         |
| com.weather.Weather             | Adid, Location                          | 110         |
| com.qiemoji.inputmethod        | Adid, IMEI, AndroidId                  | 111         |
| All                             | All                                     | 3039        |

| Domain Name                     | Leak Types                              | # Exposures |
|--------------------------------|-----------------------------------------|-------------|
| mopub.com                      | Adid                                    | 2380        |
| isnssdk                         | AndroidId, IMEI                         | 805         |
| roblox.com                     | Location                                | 679         |
| applovin.com                   | Adid                                    | 566         |
| rbxcdn.com                     | Location                                | 561         |
| appsflyer.com                  | Adid                                    | 549         |
| facebook.com                   | Adid                                    | 391         |
| bitmango.com                   | Adid                                    | 371         |
| goforandroid.com               | AndroidId                               | 262         |
| ithlhs.com                     | Adid                                    | 219         |
| pocketterms.com                | AndroidId                               | 211         |
| ksmobile.net                   | SerialNumber, Location, AndroidId       | 159         |
| tapjoy.com                     | Adid, AndroidId                         | 151         |
| tapjoyads.com                  | IMEI, AndroidId                         | 147         |
| wish.com                       | Adid                                    | 139         |
| paypal.com                     | AndroidId                               | 131         |
| All                             | All                                     | 3039        |

Figure 3: Understanding the behavior of app that expose PII through graph analysis of the AntShield dataset. The graph consists of nodes corresponding to destination domains and edges representing the similarity of two domains. Two domains are similar if there are common apps that send packets with PII exposures to both domains; the more common apps expose PII to these domains, the more similar they are, the larger the width of an edge between them. The color of a domain node indicates the types of PII it receives. One can observe from the graph structure that domains form communities that capture interesting patterns: (1) The large communities on the left and bottom consist mostly of ad networks; ad exchanges are nodes in between ad communities; (2) Facebook/Google domains are a different community on their own, on the top left; (3) small apps contact only their own domain, leading to isolate domain nodes; (4) domains in the same community receive the same type of PII (as indicated by the color of nodes). Third, not all domains belong to a community: some are well-behaved and are contacted only by their own app. For instance the white-colored domain zillow.com towards the bottom center of the graph is an isolate node and only receives information about the user’s location, which makes it similar to a well-behaved app.
Table 4: Applications with "jailbroken" field

| App Name                  | Domain               | PII Types           |
|---------------------------|----------------------|---------------------|
| com.bitstrips.imoji       | pushwoosh.com        | AndroidId           |
| 10.2.32, 10.3.76          | upsight-api.com      | Location, AndroidId |
| com.nianticlabs-pokemongo | upsight-api.com      | AndroidId           |
| com.psafe.msuite 3.11.6, 3.11.8 | onesignal.com    | AndroidId           |
| com.yelp.android 9.5.1    | onesignal.com        | AndroidId           |
| com.zeptolab.ctr.ads      | namcowireless.com    | AndroidId           |
| 2.8.0                     |                      |                     |
| com.namco.bandai-games.pacman tournaments 6.3.0 | upsight-api.com | AndroidId           |
| com.hatuuge.casino.slots 2.3.185 | pushwoosh.com | AndroidId           |
| com.cmplay.dancingline   | pushwoosh.com        | AndroidId           |

In general, similarity of apps and domains based on their PII exposure found in their network activity can be exploited to detect and prevent abusive behavior (e.g., advertising, tracking, or malware) in mobile traffic. This is one promising direction for future work.

4 User Study: Mobile Data Privacy in Context

In this section, we design a user study on Amazon Mechanical Turk (MTurk) in order to assess user’s awareness and understanding of mobile data exposure, as well as their level of concern and potential for adopting solutions. We use the datasets collected in the measurement study in the previous section to present participants with real-world scenarios of private information that was actually exposed by mobile apps in our experiments. We present the user with information about the types of PII exposed, as we as information about the context this exposure occurred, i.e., whether the PII is shared with the application or a third party server; what was the app category/intended functionality; whether it is shared in clear or encrypted text, etc. We then ask the users to assess the legitimacy (i.e., whether the information is needed for the app’s functionality) and the risk posed by the particular PII type exposed in that particular context. We also educate the participants on how a single piece of PII can lead to even more information being discovered when combined with data fetched from a data broker. Finally, we ask users before and after the survey whether they would use privacy enhancing tools. To the best of our knowledge, this user study is the first one that collects and analyzes user input in such fine granularity (context) and on actual (not just potential or permitted) privacy exposures at large scale (as found in the packet traces of the measurement study). Section 4.1 presents the design rationale and questions asked in the MTurk study. Section 4.2 summarizes and analyzes the responses from 220 participants.

4.1 User Study Design

MTurk Setup. We designed a Human Intelligence Task (HIT) on Amazon Mechanical Turk (MTurk) [1] and restricted it to workers who are based in the U.S., are at least 18 years old, have completed at least High School (in the US), and own a smartphone or a tablet device. Our study was approved by the Institutional Review Board (IRB) of our Institution (details are omitted from this double-blind version). The workers were rewarded at a rate of $0.10 per minute of their time – a standard followed in other studies [19, 22, 24]. We allotted 30 minutes for the completion of our HIT, but the majority of workers completed it within 13 minutes approximately. The participants had to pass at least one of three attention check questions in order to have their HIT approved and to receive the $3.00 payment. The HIT was open for 9 days in early May 2018. At the end, we analyzed the responses of 220 workers that passed the attention test.

Demographic questions. First, we asked a set of demographic questions, such as educational level, age, and employment sector (tech vs. non-tech). We also asked what kind of mobile OS they use and how many different apps they use daily. In addition to these questions, we added three attention-check questions to prevent workers from gaming the system by providing answers randomly. We discard answers from participants that failed to correctly answer all three attention check questions.

Categorization questions. The main goal of our study is to learn how concerned are users about privacy exposures in different contexts, defined as: the type of PII exposed, the app category, whether the information was shared with a relevant application server (thus may be useful for the functionality of the app) or third party advertiser and analytics servers, and whether it is shared in plain text or is encrypted. To that end, we first defined these terms and categories as shown in Fig. 4.

First, we asked the participants how comfortable they are with sharing various PII types with different types of remote servers, as depicted in Fig. 5. Different PII types include: various device ids (such as phone number, IMEI, IMSI, ICCID, Android Id, etc), user ids (e.g., email, Advertiser Id, username and password), location (GPS coordinates), and demographic information (e.g., gender, city, zipcode, first and last name). Destination servers are roughly divided into two categories: app developers vs. ad & analytics servers. The rationale is that the application servers may need the PII to perform their functionality (e.g., Google Maps clearly needs location) while

---

8

6We went through the IRB process in our institution and obtained exempt research registration HS# 2018-4272 “Amazon Mechanical Turk Survey on Mobile Data Privacy”
third party servers do not (thus causing more of a privacy leak rather than an exposure). For each pair of (PII type, destination type) we asked the participants to rate their comfort level of sharing that PII type with that remote server, on a scale from 0 to 3; where 0 represents the least concern and 3 represents maximum concern (and their willingness to pay for a privacy-preserving solution). See Fig. 5 for details.

Second, we asked participants to rate their concern in real-case scenarios of PII exposures from our dataset (Sec. 3). For each packet that contained a PII in our dataset, we considered a broader context beyond just PII type and destination server type (i.e., application server or ad/analytics server). We also considered the category of the app (e.g., game vs navigation), whether the PII was encrypted or sent in plain text, and the frequency of this PII being exposed by this app category. The rationale is that the same PII type exposed may be more or less concerning to users depending on the context. For example, location exposed by a navigation app in an encrypted packet is probably needed for the app to function, while sending a user id to an unrelated third party (e.g., advertiser) server, frequently and/or in plain text, is indeed a privacy leak.

A side benefit of the aforementioned categorization is that it helped reduce the number of cases to be evaluated by users. Out of 8,579 exposures total (Table 1), 1,726 are unique when considering the application responsible, the type of PII, the destination host, and level of encryption. To further reduce the number of cases to label, we grouped the applications based on their Google Play Store [5] category and destination type (ad & analytics or not). To find ad & analytics domains, we used the hpHost [6] list as it was found to be the most comprehensive list for the mobile ecosystem to date [31]. These grouping reduced the total number of unique combinations to 256, which contained 23 unique Google Play Store categories (out of 36 total). We split these combinations into five batches of HITs, where each HIT contains five (or three for the last batch) categories of apps to be labeled. To prepare the participants for the labeling task, we first showed a “warm up” question (Fig. 6(a)) with a hypothetical scenario of the Roblox app exposing certain PII and asked them to assess the risk (Fig. 4(a)). Next, we asked the participants to label the exposure scenarios for each of the five categories in their HIT – example shown in Fig. 6(b). We also provide an example app out of each category (from our actual dataset) along with a link to the app’s Google Play Store page.

Assessing User Concern and Possible Actions. In order
to assess participants’ privacy awareness and understanding, we asked another set of questions shown in Fig. 7. First, we asked if they would use an app (e.g., AntMonitor) that can prevent privacy exposures and how much would they pay for such a service (Fig. 7(b)). It was our hope that the categorization questions described previously would educate users about mobile privacy and would make them more concerned towards the end of the survey. To educate them further, we showed them that a single PII in the hands of a data broker can help create or lookup a user profile and can reveal much more information (Fig. 7(d)); the question was based on a real scenario where we fetched data from a data broker based on a person’s email address. We then asked them (if they would use a privacy app (and if they would pay for it) to protect their privacy. Finally, we asked them if they would contribute their mobile data to an app that crowdsources information, train machine learning models and prevents privacy leaks (e.g., as in Recon [32] and in [37]). In order to assess our hypothesis that users became more privacy-aware after the categorization questions and the data broker example, the same questions appeared twice during the survey: once in the middle of the survey and once at the end.

### 4.2 User Study Results

In this section, we summarize and analyze the main results of our user study. The main observation is that users seem initially confused about the severity of PII shared in different contexts, but they are interested in and are capable of being trained. Our main findings are as follows:

- **Users do not seem to understand how severe it is to share certain PII types (especially device identifiers) with either Advertisers or Developers. This may be because some of these ids, such as Android ID, IMEI, IMSI, ICCID, are difficult to understand or relate to, yet they uniquely identify the device and/or the user and hence should not be shared with remote servers.**

- **As expected, users trust Application Developers more than Advertisers & Analytics. However, some comments stated that Developers are “a bunch of hackers behind servers,” indicating possible confusion.**

- **Users also seem confused about sharing PII in plain text vs. using encryption. Sharing PII in plain text is a bad practice, since it exposes private information not only to the destination servers but also to anyone that is sniffing into the network.**

- **Towards the end of the study, most users seem to obtain...**
Consider that you run the ROBLOX app on your smartphone. ROBLOX belongs to app category GAME in Google Play. In our experiments, we noticed that the app needs your phone’s GPS location, or your location with high frequency, to both render ROBLOX scenes (“App Developer”) and to third-party Advertising and Analytics servers. This is summarized in the following table.

(a) Warm-up question with a hypothetical scenario of a particular example application – the game Roblox.

(b) Entire game category with real cases of PII types exposed and their broader context (remote server type but also app category, encrypted/plain text, and frequency).

Figure 6: Task to assess user concern about privacy exposures in context.

A much better understanding of mobile privacy. For example, several comments state that users are grateful for our short tutorial in mobile privacy. They are willing to educate themselves more in the future and to adopt data transparency and privacy tools. These are encouraging results for future work in developing privacy-enhancing technologies.

4.2.1 Summary and Demographic Info.

We collected 223 responses, in total, on our MTurk survey. Since we posted multiple HITs, each with different categorization questions (see Sec. 4.1), some users completed more than one HIT, and there were a total of 151 unique participants. Two responses were discarded due to failing the attention-check questions, and one worker was discarded due to incomplete answers. This leaves a total of 220 valid responses, which we analyze in the rest of the section. The majority of the participants were between 25 and 34 years old, held a Bachelor’s degree and were employed in a non-tech sector. 61.8% and 37.7% of the workers were Android users, and iOS users, respectively. 118 of our workers use six to ten different apps every day, 53 use between 11 and 20 apps, 41 use fewer than five apps, and only 8 use more than 20 apps.

4.2.2 Categorization Results: Privacy Exposures in Context.

In this subsection, we provide the results we obtained by processing user answers to the categorization questions (see Figures 5 and 6). We asked participants to rank the severity of PII exposures in different contexts since whether or not an app shares PII information with third-party services (data brokers) is carefully considered.

There are several third-party services (data brokers) available online that can look up your personal information based on identifiers such as your email. This information can be shared with advertisers, trackers or anyone using the services of these data brokers. Below we show you an example of us looking up one email on such a website and getting all this information back in return.

(a) How much do users care about privacy?

(b) How much would users pay to protect their privacy?

(c) Would users contribute their data to a privacy app?

(d) Do users care more after being educated about data brokers?

Figure 7: Assessing users concern and potential actions. (a) Do users care about privacy? What are they willing to do about it: (c) share their data with a crowdsourcing system (b) use and/or pay for a privacy app. And (d) do they change attitude after being educated?
exposure is considered a privacy leak depends on the context. There are four main dimensions in each exposure: (i) its destination (app developers or advertisers), (ii) category of the app responsible, (iii) level of encryption used, and (iv) PII type. These dimensions play an important role in distinguishing privacy leaks from exposures.

Figure 8(a) demonstrates the results when we asked the users how concerned they are with sharing a certain information type with a particular destination (Advertisers & Analytics or App Developer servers), as shown previously in Fig. 5. In this task, we observe that sharing all types of PII is concerning for our participants, regardless of its destination, and they are willing to use a free app to protect them. Furthermore, they would also pay for a tool that protects their phone number and password from leaking to advertisers. As expected, participants are more concerned over their precise longitude and latitude coordinates being shared as opposed to zipcode and city. Overall, our participants seem to trust developers more than advertisers, which is not surprising.

In Fig. 8(b), we also provide our own “expert” rating for comparison. In particular, we would like to protect device identifiers regardless of their destination, by using a paid solution if a free version is not available. In contrast, our participants chose only to protect their device identifiers with a free solution and they are not willing to pay to protect them. Moreover, they give similar rating to their location data regardless of the destination, although they should be more careful with advertisers than developers, as there are apps that need location in order to function. On the other hand, Advertiser ID should not be protected as well as the other identifiers, since this id is known by advertisers anyway.

Understanding whether PII is needed by the app. We ask users to assess whether sharing a particular PII is legitimate and needed for the functionality of the app (or more broadly by apps in the same category), or if it is unnecessary and potentially harmful. For example, GoogleMaps is an app in the Navigation category and needs to share location to provide the service.

Fig. 9 presents a heatmap of the perceived severity of different PII types per application category. The x-axis show the different PII types and the y-axis contains the categories of apps responsible for sending that PII over the network. The values and colors represent the level of concern the participants have regarding each pair (category, PII type): “not needed by the app and maybe harmful” (value 4 - dark color), “not needed by the app and not harmful” (value 3), “needed by the app” (value 2), “don’t care” (value 1 - light color). White color (or 0 value) represents missing combinations of PII and category, that were not present in our datasets. In order to produce the heatmap, we consider the mean values of all participants’ answers.

For comparison, we also show a heatmap labeled by us to express our “expert” opinion. First, location information (latitude and longitude coordinates, zipcode, and city) should be available to system, maps & navigation, weather, and travel & local categories only. All the other categories do not need this kind of information for their functionality and hence when apps transmit such data, it may be considered harmful. Second, user identifiers, such as email, password, username, gender, etc. may be needed in some app categories (e.g., social, games, communication), but not in others (e.g., photography, personalization, tools). Finally, device identifiers, such as IMEI, IMSI, Serial Number, ICCID, and Android ID should not be used by any app category, except perhaps for the System. These identifiers are unique and can be used to track users across different apps and build profiles, such as the one we showed in Fig 7(d). Furthermore, Google explicitly discourages app developers from using these identifiers and asks them to instead create and use their own unique identifiers within their app [4].

Comparing the two heatmaps in Fig. 9 reveals that users are not as concerned about device identifiers as they should be: 32.6% of the device identifiers per app category were categorized by workers as “Not needed by the app and not harmful,” although they should only be accessed by the System. However, other responses make sense: app categories that should not require any information to function are trusted the least (games, lifestyle, music & audio, personalization, and tools), and categories such as weather and travel & local are trusted with location information.

Understanding Encryption. We further split the heatmap in Fig. 9 into Fig. 10(a) and Fig. 10(b), depending on whether the unencrypted packet containing the PII was unencrypted or encrypted, respectively. Our heatmaps not only reveal participants’ opinions about exposures, but also showed behavioral patterns of app categories. From Fig. 10(a), we see that the weather category does not use encryption and sends Advertiser Id, Location and City to remote servers in plain text. Similarly, maps & navigation category sends Serial Number, Location, Zipcode and City information in plain text, but encrypts the Android ID and Advertiser ID. Sending PII in plain text is more harmful since this traffic can be sniffed. Unfortunately, our MTurk participants did not seem to understand the implications of transmitting data in plain text.

Understanding Destinations. We also split our heatmaps based on whether the packet is going to App Developers (Fig. 11(b)) or Advertiser & Analytics Servers (Fig. 11(a)). As expected, developers require more types of information in contrast to advertisers. On the other hand, advertisers should ideally only require the Advertiser ID, which is indeed fetched by almost every app category. However, most of our participants indicated that none of the PII need to be sent to advertisers for the app’s functionality (only 2 out of 31 values in Fig. 11(a) are below 2.5). This indicates that perhaps users don’t consider ads being served a part of the app’s functionality. In contrast, participants indicated more trust towards app developers (Fig. 11(b)), which is expected as certain app categories require PII to function correctly. For example, the following
Figure 8: How comfortable are users with sharing PII with advertisers and app developers? (a) Average rating of user responses to categorization task of Fig. 5 (b) Recommended rating by us ("experts").

Figure 9: Heatmap severity of (PII type, app category). The darkness of the color indicates the perceived severity of the PII exposure: the darkest corresponds to 4.0 ("Not needed by the app and maybe harmful"), while the lightest indicates 1.0 ("I don’t care"). Zeros represent missing values for combinations we did not have in our datasets. We compare the average ratings among users who answered that question (on the left column) vs. labeling recommended by us (on the right column).

Figure 10: Heatmaps assessing the severity of exposures in encrypted vs. unencrypted packets, as assessed by users vs. us ("experts").

Pairs of PII and categories are expected: username for applications with logins (communication, games, and social), email for communication, and location information for travel & local and maps & navigation. However, device identifiers, such as IMEI, IMSI, Serial Number and Android Id are not needed by any app category and they should not be shared with advertisers nor developers. Our participants seem to not understand this point and they labeled most of these exposures as “Not needed and not harmful,” while in fact these ids are most likely used for tracking. One interesting finding is that
participants showed more concern over their IMEI vs. other device identifiers. This indicates that they may need more education about the other device identifiers.

4.2.3 Towards A Solution: Tools Enhancing Data Transparency and Privacy.

Fig. 12(a) shows the distribution of answers to the questions of Fig. 7, which essentially ask how much users care about privacy exposures and what they are willing to do about it. The overwhelming majority of users would like more control over their information being shared, if a free option is available. Fig. 12(b) demonstrates what payment options users would prefer for a privacy app: the majority would prefer a one-time fee between $1 and $5, than a subscription model. These results show that users are not only interested in using privacy-preserving tools but are also willing to pay for them. Towards the end of the study, after being educated about the extent and severity of sharing PII, more users were willing to pay for a privacy app, e.g., compare Fig. 12(a) to Fig. 12(c). This demonstrates that although users may originally not be aware or understand the risks of privacy exposures, they become more weary after being educated about the occurrence and risks of sharing PIIs (especially after learning the power of data brokers). Finally, Fig. 12(d) shows the user’s willingness to contribute data to a privacy-preserving tools that crowd-source information (such as Lumen, Recon, AntMonitor) before and after completing our survey. Once again, completing the study made users more open towards using and helping a privacy-preserving app.

4.2.4 User Comments.

At the end of the study, we asked participants for their comments and suggestions; 99 out of 220 users provided their feedback. In Fig. 13, we summarize the comments of all workers in the form of a wordcloud. Overall, our MTurk workers seemed satisfied with the survey and stated that it was an educational experience. Most of the participants are thankful for the opportunity to contribute their data and are interested in the possibility of using privacy-preserving tools in the future.
for our short tutorial on mobile privacy, as they gained more knowledge about privacy and how different apps share their information with various destinations. At the end of the survey, they seem more concerned about privacy and interested in a solution, including a privacy app free or paid; see Fig. 12. Several participants mentioned the recent Facebook and Cambridge Analytica scandal and wondered if our user survey was inspired by it (it was not!). Below, we provide a sample of representative user comments, out of 99 total comments, grouped by two recurrent topics.

They learned a lot:

“I liked this. Was this all due to what is going on with Facebook and other privacy concerns?”

“I enjoyed taking this survey. I would not share my information with anyone that would use the information in anyway that would be damaging to my life. Right now I am getting phone calls I don’t want. I would pay to have those stopped.”

“I definitely became more concerned about how much data is taken. You don’t realize how much of your personal data is sent to advertisers and it makes you more weary of downloading and using certain apps.”

“I thought it was quite enlightening. I will certainly be paying more attention to what and how an app uses my data in the future.”

“I definitely became more concerned about my privacy. Thank you for the wake up call.”

“Good Study. Made me really think about how much of everybody’s information is really out there.”

“I am surprised to learn how much information could be linked to my email address. It is like telemarketing on steroids and I would be willing to keep that info private I think we should always have the option to keep our private info private.”

“I have become a little more concerned about my privacy as some of this was new information to me. I will definitely be doing more research on this subject in the near future. Thank you for the informative study! Everything was very clear and straightforward, I appreciate the opportunity to participate.”

They are interested in using a privacy-enhancing solution:

“Thank you. I learned a lot from this study. I hope you are able to develop something that can help protect consumers and still allow developers to flourish.”

“I would love a reliable app that limited/reduced data collection and increased privacy. Problem is that it’s not possible to know which privacy apps are reliable and effective, while still allowing a service or app to be used. Even a reliable, mainstream email alternative would be good!”

5 Conclusion & Future Directions

We provided a combination of a measurement and a user study of actual PII exposed by mobile apps. We also defined and analyzed the context (PII type, destination domain, app category/functionality, background/foreground, use of encryption vs. plain text) where these PII exposures occur, and we distinguished between PII exposure and PII leak (which is more likely to be harmful) depending on the context. In the measurement study, we collected and analyzed a new richer dataset, which reveals interesting PII exposures and patterns, some of which were previously unknown. Preliminary graph analysis revealed interesting patterns of apps and domains colluding to expose private information. In the user study,
we compiled the large amount of information from the measurement study into a smaller number of categories (contexts) and asked users to assess the privacy exposures in the actual context they appeared, w.r.t. their legitimacy and risk they pose. Most users were initially unaware of the severity and potential implications of PII exposures: they could not identify the most critical PII or context (e.g., that it may be ok to share information with the application server instead of third parties, such as advertisements and trackers). However, they appeared to appreciate the information they got through the study, which made them more willing to adopt privacy enhancing tools. Our analysis combines the scale and coverage of the network-based measurement study with the fine-granularity user input assessing privacy exposures in context.

There are several directions for future work, building on the observations of this paper. First, the behavioral analysis of PII leaks at the end of Section 3 revealed similarities in the way apps and destination domains extract PII from mobile devices. Those observations can be further exploited to design machine learning approaches that can detect packets with potential privacy exposures [11, 34], further inspect them and eventually prevent an actual leak (e.g., by using real-time tools like AntMonitor to block a packet or obfuscate a PII). Second, the user study showed that users are interested in and are capable of being educated about their data, and they want to adopt better privacy practices and new tools (such as AntMonitor, Recon or Lumen) to enhance data transparency and privacy control.

6 Acknowledgments

We would like to thank Milad Asgari Mehrabadi for providing figures 2 and 3 for this paper.

This work has been supported by NSF Awards 1649372, 1815666, 1900654 and by a UCI Proof-of-Concept Award in

We would like to thank Milad Asgari Mehrabadi for providing way apps and destination domains extract PII from mobile

Monitor

References

[1] Amazon mechanical turk. https://www.mturk.com.

[2] Antmonitor project. https://athinagroup.eng.uci.edu/projects/antmonitor/.

[3] App annie. https://www.appannie.com.

[4] Best Practices for Unique Identifiers. https://developer.android.com/training/articles/user-data-ids.

[5] Google play. https://play.google.com/store?hl=en.

[6] hpHosts - Ad and Tracking servers only. https://hosts-file.net/ad_servers.txt.

[7] PhoneLab, University at Buffalo. https://www.phonelab.org/.

[8] Speed guide: Ports database. http://www.speedguide.net/ports.php.

[9] Ui/application exerciser monkey. https://developer.android.com/studio/test/monkey.html.

[10] Hazim Almuhimedi, Florian Schaub, Norman Sadeh, Idris Adjerid, Alessandro Acquisti, Joshua Gluck, Lorrie Faith Cranor, and Yuvraj Agarwal. Your Location has been Shared 5,398 Times!: A Field Study on Mobile App Privacy Nudging. In Proceedings of the 33rd annual ACM conference on human factors in computing systems, pages 787–796. ACM, 2015.

[11] Evita Bakopoulou, Balint Tillman, and Athina Markopoulou. A federated learning approach for mobile packet classification, 2019.

[12] M. Bastian, S. Heymann, M. Jacomy, et al. Gephi: an open source software for exploring and manipulating networks. Icwsm, 8, 2009.

[13] Saksham Chitkara, Nishad Gothoskar, Suhas Harish, Jason I Hong, and Yuvraj Agarwal. Does this app really need my location? context-aware privacy management for smartphones. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(3):42, 2017.

[14] Andrea Continella, Yanick Fratantonio, Martina Lindorfer, Alessandro Puccetti, Ali Zand, Christopher Kruegel, and Giovanni Vigna. Obfuscation-resilient privacy leak detection for mobile apps through differential analysis. 2017.

[15] M. Egele, C. Kruegel, E. Kirda, and G. Vigna. Pios: Detecting privacy leaks in ios applications. In NDSS, 2011.

[16] W. Enck, P. Gilbert, B. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM TOCS, 2014.

[17] H. Falaki, D. Lymberopoulos, R. Mahajan, S. Kandula, and D. Estrin. A First Look at Traffic on Smartphones. In Proc. of the 10th ACM SIGCOMM Conf. on Internet Measurement, Melbourne, Australia, Nov. 2010.

[18] C. Gibler, J. Crussel, J. Erickson, and H. Chen. AndroidLeaks: Automatically detecting potential privacy leaks in android applications on a large scale. In TRUST, 2012.
[19] Qatrunnada Ismail, Tousif Ahmed, Kelly Caine, Apu Kapadia, and Michael Reiter. To Permit or Not to Permit, That is the Usability Question: Crowdsourcing Mobile Apps’ Privacy Permission Settings. Proceedings on Privacy Enhancing Technologies, 4(4):118–136, 2017.

[20] Zach Jorgensen, Jing Chen, Christopher S. Gates, Ninghui Li, Robert W. Proctor, and Ting Yu. Dimensions of Risk in Mobile Applications. Proceedings of the 5th ACM Conference on Data and Application Security and Privacy - CODASPY ’15, pages 49–60, 2015.

[21] Yuanchun Li, Fanglin Chen, Toby Jia-Jun Li, Yao Guo, Gang Huang, Matthew Fredrikson, Yuvraj Agarwal, and Jason I. Hong. Privacystreams: Enabling transparency in personal data processing for mobile apps. Proc. ACM Interact. Mob. Wearable Ubiquitous Technol., 1(3):76:1–76:26, September 2017.

[22] Jialiu Lin, Norman Sadeh, Shahriyar Amini, Janne Lindqvist, Jason I. Hong, and Joy Zhang. Expectation and purpose: Understanding Users’ Mental Models of Mobile App Privacy through Crowdsourcing. Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp ’12, page 501, 2012.

[23] Bin Liu, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun (Aerin) Zhang, Norman Sadeh, Yuvraj Agarwal, Alessandro Acquisti, Mads Schaarup Andersen, Florian Schaub, Hazim Almuhimedi, Shikun (Aerin) Zhang, Norman Sadeh, Yuvraj Agarwal, and Alessandro Acquisti. Follow My Recommendations: A Personalized Privacy Assistant for Mobile App Permissions. Twelfth Symposium on Usable Privacy and Security (SOUPS 2016), (Soups):27–41, 2016.

[24] Winter Mason and Siddharth Suri. Conducting Behavioral Research on Amazon’s Mechanical Turk. Behavior Research Methods, 5(5):1–23, 2010.

[25] Abhinav Mehrotra, Sandrine R. Müller, Gabriella M. Harari, Samuel D. Gosling, Cecilia Mascolo, Mirco Musolesi, and Peter J. Rentfrow. Understanding the Role of Places and Activities on Mobile Phone Interaction and Usage Patterns. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(3):1–22, 2017.

[26] Yuhong Nan, Zhemin Yang, Xiaofeng Wang, Yuan Zhang, Donglai Zhu, and Min Yang. Finding Clues for Your Secrets: Semantics-Driven, Learning-Based Privacy Discovery in Mobile Apps. 2018.

[27] A. Rao, A. Molavi Kakhki, A. Razaghpanah, A. Tang, S. Wang, J. Sherry, P. Gill, A. Krishnamurthy, A. Legout, A. Mislove, and D. Choffnes. Using the Middle to Meddle with Mobile. Technical report, Northeastern University, Dec. 2013.

[28] Ashwini Rao, Florian Schaub, Norman Sadeh, Alessandro Acquisti, and Ruogu Kang. Expecting the Unexpected: Understanding Mismatched Privacy Expectations Online. the Proceedings of the Twelfth Symposium on Usable Privacy and Security (SOUPS 2016), (Soups):77–96, 2016.

[29] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman, and V. Paxson. Haystack: In Situ Mobile Traffic Analysis in User Space. arXiv:1510.01419, Oct. 2015.

[30] A. Razaghpanah, N. Vallina-Rodriguez, S. Sundaresan, C. Kreibich, P. Gill, M. Allman, and V. Paxson. Haystack: A multi-purpose mobile vantage point in user space. arXiv:1510.01419v3, Oct. 2016.

[31] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, and Phillipa Gill. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. 2018.

[32] J. Ren, A. Rao, M. Lindorfer, A. Legout, and D. Choffnes. Recon: Revealing and controlling pii leaks in mobile network traffic. In In ACM MobiSys), volume 16, 2016.

[33] Jingjing Ren, Martina Lindorfer, Daniel J Dubois, Ashwin Rao, David Choffnes, and Narseo Vallina-Rodriguez. Bug Fixes, Improvements, ... and Privacy Leaks A Longitudinal Study of PII Leaks Across Android App Versions. (February), 2018.

[34] A. Shuba, E. Bakopoulou, and A. Markopoulou. Privacy leak classification on mobile devices. In 2018 IEEE 19th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pages 1–5, 2018.

[35] A. Shuba, A. Le, E. Alimpertis, M. Gjoka, and A. Markopoulou. Antmonitor: System and applications. arXiv:1611.04268, 2016.

[36] A. Shuba, A. Le, M. Gjoka, J. Varmarken, S. Langhoff, and A. Markopoulou. Antmonitor: Network traffic monitoring and real-time prevention of privacy leaks in mobile devices. In ACM Mobicon Demo and Short Paper (and best demo in S3), September 2015.

[37] Anastasia Shuba, Evita Bakopoulou, Milad Asgari Mehrabadi, Hieu Le, David Choffnes, and Athina Markopoulou. Antshield: On-device detection of personal information exposure. arXiv preprint arXiv:1803.01261, 2018.
[38] Y. Song and U. Hengartner. PrivacyGuard: A VPN-based Platform to Detect Information Leakage on Android Devices. In Proc. of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, Oct. 2015.

[39] Yihang Song. Privacyguard: A vpn-based approach to detect privacy leakages on android devices. 2015.

[40] Lei Tian, Shaosong Li, Junho Ahn, David Chu, Richard Han, Qin Lv, and Shivakant Mishra. Understanding user behavior at scale in a mobile video chat application. Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing - UbiComp '13, page 647, 2013.

[41] N. Vallina-Rodriguez, A. Auçinas, M. Almeida, Y. Grunenberger, K. Papagiannaki, and J. Crowcroft. RILAnalyzer: A Comprehensive 3G Monitor on Your Phone. In Proc. of IMC, Barcelona, Spain, Oct. 2013.

[42] Max Van Kleek, Ilaria Liccardi, Reuben Binns, Jun Zhao, Daniel J Weitzner, and Nigel Shadbolt. Better the devil you know: Exposing the data sharing practices of smartphone apps. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pages 5208–5220. ACM, 2017.

[43] Haoyu Wang, Yuanchun Li, Yao Guo, and Yuvraj Agarwal. Understanding the Purpose of Permission Use in Mobile Apps. ACM Trans. Inf. Syst. Article, 35(43), 2017.

[44] Na Wang, Bo Zhang, Bin Liu, and Hongxia Jin. Investigating Effects of Control and Ads Awareness on Android Users' Privacy Behaviors and Perceptions. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services, pages 373–382. ACM, 2015.

[45] Yichuan Wang and David Chu. EarlyBird: Mobile Prefetching of Social Network Feeds via Content Preference Mining and Usage Pattern Analysis Categories and Subject Descriptors. MobiHoc, pages 67–76, 2015.

[46] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos. ProfileDroid: Multi-Layer Profiling of Android Applications. In ACM MobiCom, Aug. 2012.

[47] Qiang Xu, Jeffrey Erman, Alexandre Gerber, Zhuoqing Mao, Jeffrey Pang, and Shobha Venkataraman. Identifying diverse usage behaviors of smartphone apps. In Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference, IMC '11, pages 329–344, New York, NY, USA, 2011. ACM.