嗨Wency，尽管本文研究了移动应用程序的“侵入性”，但我认为它与运动状态的变化有关，因为移动应用程序的“侵入性”活动会导致用户将其当前活动更改为“侵入性”移动应用程序所提示的活动。
例如，在注释＃10上，“想法”状态与“中断”之间存在有趣的关系。
在注释＃13上，有用于可视化使用因素和上下文因素的说明。
您会在这里找到有用的类别的移动应用程序，它们会导致用户活动和动作状态发生变化。
量化“侵入性”的过程也可以考虑量化“运动状态”。

标题：不足为奇：通过检测客观上下文偏差来衡量智能手机应用程序的侵入性
文件：No_surprises_measuring_intrusiveness_annotated.pdf
资源: https://dl.acm.org/citation.cfm?id=2517864

注解
这些项目符号上的数字对应于添加到此研究文档中的数字。

1. 具体来说，我们发明了一个框架，用于基于智能手机应用程序的数据访问行为来定性评估和定量测量智能手机应用程序的“侵入性”。
2. 一个新的“侵入性评分”，基于对来自4个应用类别的33个流行Android应用的经验测试收集的“真实数据访问”，以数字方式衡量了上下文外数据的收集。
3. 我们的研究还表明，“侵入性得分”有助于比较表现出相似数据访问类型的应用。
4. 我们通过对照智能手机上的信息规范检查应用程序的数据访问行为，并使用它们来量化“侵入性”，从而提高了移动应用程序的透明度。
5. 目的是通过监视不同用户上下文中的数据访问来衡量和比较每个应用程序的“侵入性”级别。我们分析了从以下4种Android应用类别（分别为游戏（10个应用），消息传递（8个应用），摄影（6个应用）和社交应用）中列出的前20个应用2中的6–10个应用的实时测试数据获得的数据访问模式。 （9个应用程序），总共33个应用程序。
6. 我们分配了两名熟悉该研究的研究助理，以在4种预定义的与隐私相关的使用情况下测试33个应用程序：1）用户闲置，移动。 2）用户闲置，用户不动。 3）用户活跃，移动。 4）用户活跃，不动。
7. 为了量化“侵入性”，我们必须首先确定一些导致每个数据访问的“侵入性”的因素。我们选择关注两个主要因素：1）数据类型可识别性和2）使用因素。数据类型可识别性可衡量与特定类型的个人数据（例如GPS位置，短信，联系电子邮件）相关的固有敏感性。使用因素根据触发访问的条件来衡量访问的“侵入性”。
8. 在表1中，GPS位置比小区位置（3）得分更高（5），这是因为GPS传感器提供了更精确的位置信息来跟踪用户移动性模式，而不仅仅是依赖于所连接的蜂窝小区的位置区域代码。
9. 根据对一些流行应用程序的初步测试，我们发现某些数据访问是由特定的使用模式触发的，即：1）用户的活动状态（活动或空闲），以及2）当时用户的移动状态（移动/不移动）。

10. 如果某个应用程序正在访问用户的位置而其活动相对于该应用程序处于空闲状态（使用系数1），则我们认为这是高度“侵入式”的数据访问。这是因为在用户闲置期间发生的访问很可能是意外的。
11. 一个“上下文包”模型，该模型计算给定应用执行的总敏感访问中每个上下文的频率。我们还为每个访问上下文分配“侵入性”权重，并使用比例频率（基于每个上下文计数）来标准化每个访问上下文对应用程序整体“侵入性”的贡献。图2概述了从AppWindow为智能手机上安装的所有应用收集的原始测试数据开始，为每个应用生成上下文包的过程。
12. 当前台应用与请求应用匹配且“屏幕模式”开启时，用户的活动状态为“活动”。如果用户的上下文经度或纬度与AppWindow记录的先前访问所收集的上下文不同，则其运动状态为“正在运动”。数据类型，移动状态和活动状态共同构成了我们所谓的访问上下文。
13. 为了生成给定应用程序的视觉效果，我们制作一个表示其“上下文包”的图形，对（1）访问上下文列表进行排序，方法是在垂直轴上增加“侵入性”（2）每个上下文的百分比贡献（由其频率计数确定）在水平轴上为“指纹”频段（请参见图3）。
14. 首先，我们列出所有可能的访问上下文。通过计算使用因素和数据类型的所有可能组合来生成访问上下文列表。我们将这些因素称为上下文因素，包括：1）数据类型及其相应的数据可识别性权重（19个值），2）用户的应用活动状态（2个值）和3）用户的移动状态（2个值）。
15. 结合每个因素，我们总共获得76个独特的环境（19 x 2 x 2）。然后，我们将琐碎的上下文从考虑中删除。具体来说，不会识别位置的数据类型（短信，联系人，电话号码，电子邮件，音频，照片，设备ID，SIM序列号，“ msisidn”，语音邮件，浏览，“ imsi”，传感器加速度计，传感器旋转）将永远不会由于用户的移动而被触发。从上下文列表中删除它们后，剩下48个剩余上下文。
16. 为了确定“侵入性”的相对排名，我们通过对步骤1中列出的3个上下文因素中的每个上下文的48个上下文列表进行排序来对上下文进行排名，顺序如下：1）用户的应用活动状态（空闲前处于活动状态）， 2）数据类型“可识别性”（升序），3）用户的移动状态（移动前不移动）。
17. “侵入式公式”输出应用的整体“侵入式得分”。
18. 我们将“侵入性公式”应用于在实验研究中测试的33个应用程序，并在本节中报告其“侵入性得分”和“侵入性子分数”。在表2中，我们报告了4个应用类别中每个类别的平均“侵入性分数”，并基于这些分数提供了相对排名，以便获得跨应用类别的广泛“侵入性”感。
19. 在用户闲置期间发生的访问活动所占比例较大，导致“消息传递”和“社交应用程序”具有更高的“侵入性得分”。
20. 由于敏感数据（短信，照片，音频，联系人，电子邮件，GPS位置，设备ID，“ imsi”，“ msisdn”，wifi）的请求种类繁多，因此消息传递应用在“侵入性”类别中排名最高用户空闲期间的访问活动。

21. 每个应用程序都有一个唯一的指纹，可以唯一地量化其数据访问行为。访问行为的这些差异会影响相对的“侵入性”。
22. 空闲活动的级别有时对相对“侵入性”的影响要大于所访问数据的类型。对于访问类似数据类型的应用程序（例如同一类别中的应用程序）尤其如此。因此，闲置活动的水平是评估“侵入性”的高度相关因素。

No Surprises: Measuring Intrusiveness of Smartphone Applications By Detecting Objective Context Deviations

Frances Zhang
MIT Computer Science and Artificial Intelligence Lab
Cambridge, MA
frango@mit.edu

Fuming Shih
MIT Computer Science and Artificial Intelligence Lab
Cambridge, MA
fuming@mit.edu

Daniel Weitzner
MIT Computer Science and Artificial Intelligence Lab
Cambridge, MA
djweitzner@csail.mit.edu

ABSTRACT

We address the challenge of improving transparency for smartphone applications by creating tools that assesses privacy risk. Specifically, we invented a framework for qualitatively assessing and quantitatively measuring the intrusiveness of smartphone applications based on their data access behaviors. Our framework has two essential components. The first component is the Privacy Fingerprint, a novel visualization that is concise yet holistic. It captures each app's unique access patterns to sensitive personal data, including which types of behaviors and under which privacy-relevant usage contexts the data are collected. The second component is a new Intrusiveness Score that numerically measures out-of-context data collection, based on real data accesses gathered from empirical testing on 33 popular Android apps across 4 app categories. Specific attention is paid to the proportion of data accesses that occurs while the user is idle, raising the perceived level of intrusiveness and exposing the profiling potential of an app. Together, these components will help smartphone users decide whether to install an app because they will be able to easily and accurately assess the relative intrusiveness of apps. Our study also demonstrates that the Intrusiveness Score is helpful to compare apps that exhibit similar types of data accesses.

Categories and Subject Descriptors

K.4 [**Computers and Society**]: Privacy; D.2.4 [**Software Engineering**]: Software/Program Verification—*Statistical methods*

Keywords

privacy; context; intrusiveness; visualization; access pattern

WPES'13, November 4, 2013, Berlin, Germany.

http://dx.doi.org/10.1145/2517840.2517864 .

1. INTRODUCTION

In today's world, where data is the new currency and mobile phones the new vehicle for information exchange, more and more personal information is placed onto mobile devices, providing apps and third party advertisers with a wealth of sensitive information to mine and profile user behavior.

With privacy breaches publicly reported every few days[1], smartphone users grow wary over the privacy risk posed by mobile apps which seek to access personal information. A recent Pew survey found that 54% of smartphone users have avoided installing an app after they discovered how much personal information they had to share in order to use it [9]. When choosing which app to install, there is a compelling need for mechanisms that clearly and concisely help users gain a holistic and accurate understanding about the sensitive access behaviors of apps they may use.

Our work is motivated by the need for a more effective mechanism that will help users make informed decisions about choosing "privacy-friendly" apps. Because we cannot yet consistently predict what the collected privacy-sensitive information will be used for and whom the apps will share it with, we cannot measure the direct harm created by the privacy intrusion. Instead, we improve mobile app transparency by examining the apps' data access behaviors against information norms on the smartphone and use them to quantify *intrusiveness*. We characterize perceived intrusiveness as the "out-of-context" actions of gathering personal information that result in a sense of "privacy loss" [6]. For example, an app is considered more intrusive if it reads a person's information at times that are inconsistent with the person's expectations [12]. Specifically, we integrate two main privacy concepts: identifiability of the requested data [8] and appropriateness of the context [15], as the central constructs for quantifying the intrusiveness of an app.

We formalize a framework that quantifies the intrusiveness of sensitive data accesses performed by each app. We quantify this subjective feeling based on more objective indicators gleaned from an app's list of data accesses, namely: 1) degree of personal identifiability of the data accessed (e.g., sms is more personally identifiable than a single gps location), and 2) the privacy-relevant usage contexts under which the sensitive data are obtained (e.g., the user is not using the app, but his/her location is continuously being collected). The ultimate goal of this work is to provide smartphone users with a quantifiable metric which they can use to easily and effectively assess and compare the relative intrusiveness of apps.

We believe that measuring intrusiveness is a pragmatic methodology for measuring the privacy risk caused by app usage, as the level of intrusiveness can be reduced with self-regulatory behavior, as demonstrated by contextually appropriate data gathering. We hope that the availability of this measurement will prompt app developers to clarify the scope of their app's "context of interaction", as recommended in the recent FTC report [4], and that this context will align well with users' expectations of privacy.

[1] `http://bit.ly/privacybreachnews`

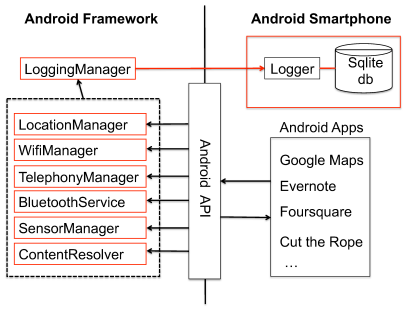

Figure 1: AppWindow Architecture

The intrusiveness framework is built incrementally, with each step motivating a different part of our work: (1) We determine the types and patterns of data accesses performed by smartphone apps by building our own context-aware information monitoring system called AppWindow. (2) We collect real access data from apps by performing an experimental study that tests 33 popular Android apps across 4 app categories. (3) We process the data to generate concise and visual representations (Privacy Fingerprint, Intrusiveness Score) of intrusiveness to simplify and aid the privacy assessment experience.

Our work makes the following contributions: (1) We develop a framework for measuring privacy risk by *quantifying* intrusiveness, whereas current transparency mechanisms focus largely on *identifying* sensitive datatypes. (2) Our privacy assessment framework accounts for the intrusiveness of privacy-relevant *usage contexts* of real data accesses, not just the *type* of data requested. (3) We use our intrusiveness framework to propose a novel transparency interface that is more concise and easy to understand.

The rest of the paper is organized as follows. In Section 2, we present the implementation details behind AppWindow, which collects the intrusiveness data. We present the study capturing apps' Intrusiveness Scores and discuss the design of the intrusiveness framework in Section 3. Section 4 shows the resulting intrusiveness scores with discussions in Section 5. Section 6 discusses the related work. Finally, we conclude our work in Section 7 .

2. IMPLEMENTATION

We implemented an information monitoring framework, App Window, to track all sensitive data requests made by Android apps. AppWindow is composed of two major components: the **Monitoring Framework** and the **Logger Application**. Figure 1 shows how the Monitoring Framework (left of vertical line) and Logger App (right of vertical line) fit together in the overall architecture of AppWindow. Components outlined in red represent structures that are either a modified version of existing Android Source Code or newly created for AppWindow.

The current version of AppWindow's Monitoring Framework is a light modification of existing Android framework source code, version 2.3.4_r1 (Gingerbread). To monitor the sensitive data accessed by apps, we placed additional logging code into public APIs of various Android framework modules responsible for managing potentially privacy sensitive information. For example, every time an app calls a method in LocationManager to get the GPS coordinates of the device, its actions will be logged. The LoggingManager packages up the *type* and *content* of sensitive data accessed, along with the requesting app's name into an Android Intent and sends it to the dedicated Logger Application residing on the client device.

The Logger App is an Android application we built to store the sensitive data accesses in a dedicated SQLite database on the smartphone. The Logger App logs a transaction record for each privacy sensitive request made by any app installed on the smartphone. This transaction record includes: 1) name of requesting application, 2) datatype requested, 3) time of request, 4) contextual screen mode (on/off), 5) contextual app status (foreground/background request), and 6) contextual location.

3. QUANTIFYING INTRUSIVENESS

In this section, we discuss the design of our intrusivessness study and the approach for quantifying intrusiveness. The goal is to measure and compare each app's level of intrusiveness by monitoring its data accesses in different user contexts. We analyzed the data access patterns obtained from realtime testing data of 6-10 apps in the top 20 apps[2] listed for each of 4 Android app categories: Games (10 apps), Messaging (8 apps), Photography (6 apps), and Social (9 apps), for a total of 33 apps[3]. We assigned two research assistants familiar with the study to test the 33 apps under 4 predefined **privacy-relevant usage contexts**: 1) User idle, user moving. 2) User idle, user not moving. 3) User active, moving. 4) User active, not moving.

For the "user idle" usage contexts (1, 2), the experimenter clicks on the app so that it appears in the foreground and then turns the screen off, representing the idle state. For the "user active" usage contexts (3, 4), the experimenter explores all features that could activate the permissions listed in the app's Android Permission List. For the "moving" usage contexts (1, 3), we require the experimenters to move around while using the app. Inversely, they must stay in one location for tests that are conducted under the "not moving" contexts (2, 4). For each app, experimenters spent a minimum of 5 minutes testing out features (some took longer than others, e.g., Games apps) and a minimum of 10 minutes testing idle times. We extend the time period for idle testing based on the assumption that apps make sparser requests when the user is not using the app in order to save battery life.

3.1 Intrusiveness Factors

In order to quantify intrusiveness, we must first identify some factors that contribute to the intrusiveness of each data access. We choose to focus on two main factors: 1) datatype identifiability and 2) usage factors. Whereas datatype identifiability measures the inherent sensitivity associated with a particular type of personal data (e.g., gps location, sms, contact email), usage factors measure the intrusiveness of an access based on the condition that triggered it.

We define datatype identifiability to represent how easily a particular type of data can be used to determine a person's identity. This term casts a broader net than Personally Identifiable Information (PII) [14]. Whereas PII only includes data that are directly linkable to personal identity, there are other types of data (e.g., gps location, wifi, bluetooth) [16] that can be useful in ascertaining a person's identity if collected over a period of time and examined in aggregate. For example, in Table 1, *gps location* has a higher score (5) than *cell location* (3) because the GPS sensor gives more precise location information for tracking the user mobility pattern than just depending on the location area code of the connected cellular cell.

Based on preliminary testing with a handful of popular apps, we found that certain data accesses are triggered by particular usage patterns, namely: 1) the user's activity status (active or idle), and 2) the user's movement status at the time (moving/not moving). These usage factors are relevant to calculating intrusiveness because accessing personal information under certain usage conditions evoke different levels of negative privacy sentiment. If an app is accessing a user's location while her activity with respect to the app is idle (usage factor 1), then we deem that to be a highly intrusive data access. This is because accesses occurring during a user's idle period are most likely unexpected.

[2]as of the ranking in August 2012
[3]For apps' information: `http://bit.ly/appwindowdata`

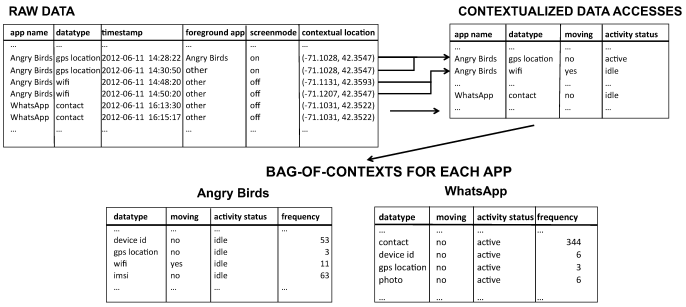

Figure 2: Process of generating a bag-of-contexts

3.2 "Bag-of-Contexts" Model

We adapt the "bag of words" model used in text processing (for computing the frequency distribution of words) to construct an analogous model for assessing the distribution of privacy-relevant usage contexts of sensitive data accesses performed by apps. Specifically, we propose a "bag-of-contexts" model that counts the frequency of each context in the total sensitive accesses performed by a given app. We also assign an intrusiveness weight to each access context and use the proportional frequency (based on each context count) to normalize the contribution of each access context to the overall intrusiveness of the app. Figure 2 outlines the process of generating a bag-of-contexts for each app, starting from raw testing data gathered by AppWindow for all apps installed on a smartphone.

First, we determine the usage context for each data access. Specifically, the user's activity status is "active" when the foreground app matches the requesting app and the "screen mode" is on. A user's movement status is "moving" if either of its contextual longitude or latitude is different from those collected for the previous access logged by AppWindow. Together, a datatype, movement status, and activity status form what we call an *access context*. Second, we count the frequency of each access context appearing in all data accesses logged by AppWindow for each app.

3.3 Formulating the Privacy Fingerprint

To generate the Privacy Fingerprint visual for a given app, we make a graph representing its bag-of-contexts, ordering the list of access contexts by increasing intrusiveness on the vertical axis and the percent contribution of each context (determined by its frequency count) as a "fingerprint" band on the horizontal axis (see Figure 3). First, we list all possible access contexts. The list of access contexts are generated by computing all possible combinations of **usage factors** and **datatypes**. We call each of these contributing factors **context factors**, including: 1) datatype and its corresponding data identifiability weights (19 values), 2) user's app activity status (2 values), and 3) user's movement status (2 values).

Combining each of the factors, we get a total of 76 unique contexts (19 x 2 x 2). We then remove trivial contexts from consideration. Specifically, datatypes that are not location-aware (sms, contact, phone number, email, audio, photo, device id, sim serial number, msisidn, voicemail, browsing, imsi, sensor accelerometer, sensor rotation) will never be triggered due to a user's movement. Taking these off the context list, we are left with 48 remaining contexts.

We then assign intrusiveness weights by ranking the contexts (listed in Table 1) in terms of increasing intrusiveness. To determine a relative ranking of intrusiveness, we rank the contexts by sorting the list of 48 contexts on each of the 3 context factors listed in Step 1, in the following order: 1) user's app activity status (active before idle), 2) datatype identifiability (in ascending order), 3) user's movement status (not moving before moving).

Table 1: Intrusiveness Weights for Access Contexts

Context ID	Datatype	Identifiability	Moving	User Activity	Context Weight
1	sms	10	no	idle	22
2	audio	10	no	idle	22
3	photo	10	no	idle	22
4	contact	10	no	idle	22
5	email	10	no	idle	22
6	phone number	10	no	idle	22
7	browsing	9	no	idle	21
8	voicemail	9	no	idle	21
9	gps location	5	yes	idle	20
10	gps location	5	no	idle	19
11	device id	5	no	idle	19
12	sim serial	5	no	idle	19
13	imsi	5	no	idle	19
14	msisdn	5	no	idle	19
15	wifi	4	yes	idle	18
16	wifi	4	no	idle	17
17	cell location	3	yes	idle	16
18	cell location	3	no	idle	15
19	neighboring cell	2	yes	idle	14
20	ip address	2	yes	idle	14
21	neighboring cell	2	no	idle	13
22	ip address	2	no	idle	13
23	sensor acce	1	yes	idle	12
24	sensor rot	1	yes	idle	12
25	sms	10	no	active	11
26	audio	10	no	active	11
27	photo	10	no	active	11
28	contact	10	no	active	11
29	email	10	no	active	11
30	phone number	10	no	active	11
31	browsing	9	no	active	10
32	voicemail	9	no	active	10
33	gps location	5	yes	active	9
34	gps location	5	no	active	8
35	device id	5	no	active	8
36	sim serial	5	no	active	8
37	imsi	5	no	active	8
38	msisdn	5	no	active	8
39	wifi	4	yes	active	7
40	wifi	4	no	active	6
41	cell location	3	yes	active	5
42	cell location	3	no	active	4
43	neighboring cell	2	yes	active	3
44	ip address	2	yes	active	3
45	neighboring cell	2	no	active	2
46	ip address	2	no	active	2
47	sensor acce	1	yes	active	1
48	sensor rot	1	yes	active	1

With this sorting order, we generate a newly ranked list of contexts, assigning an intrusiveness weight of 1 to the least intrusive context (at the bottom of the list) and incrementing by one every time we encounter a new combination of context factors as we move up the list. The final context weighting is shown in Table 1.

3.4 Computing the Intrusiveness Score

From a high-level perspective, we input an app's bag-of-contexts into a function, called the Intrusiveness Formula, and it outputs the app's overall Intrusiveness Score. The detailed of the formula is presented in the Equation 1 below:

$$x = i * \sum_{n=1}^{k/2} w_n * p_n + a * \sum_{n=k/2+1}^{k} w_n * p_n \quad (1)$$

Variables

x = an app's overall intrusiveness

i = fraction of all data accesses that were conducted during the user's idle period

a = fraction of all data accesses that were conducted during the user's active period

k = total number of access contexts (in this study, $k = 48$)

w_n = intrusiveness weight assigned to the n^{th} context

p_n = the fraction of an app's total data accesses belonging to the n^{th} access context

In words, the Intrusiveness Score is a sum of the normalized Idle Intrusiveness Subscore ($i * \sum_{n=1}^{24} w_n * p_n$) and normalized Active Intrusiveness Subscore ($a * \sum_{n=25}^{48} w_n * p_n$). As the name suggests, the Idle Intrusiveness Subscore is computed by summing the intrusiveness weights of all access contexts corresponding to a user's idle state, normalizing each weight w_n by its context's proportional frequency p_n. The Active Intrusiveness Subscore is computed similarly for all access contexts corresponding to a user's active state. The Idle Intrusiveness Subscore

Table 2: Ranking of App Categories

Ranking	App Category	Avg. Intrusiveness	σ
1	Messaging	14.89	3.43
2	Social	13.46	3.54
3	Games	8.91	4.64
4	Photography	5.81	5.05

$(\sum_{n=1}^{24} w_n * p_n)$ is normalized by the proportion frequency of all idle data accesses i, and the Active Intrusiveness Subscore $(\sum_{n=25}^{48} w_n * p_n)$ by the proportion frequency of all active data accesses a. The two normalized Subscores are added to arrive at overall Intrusiveness Score for the app. The highest possible score for an app is 22. While a single Intrusiveness Score is more concise and easier to interpret for smartphone users, we break down this score into 2 subscores – Idle Intrusiveness Subscore and Active Intrusiveness Subscore – to highlight the substantial amounts of data accesses happened when smartphone users are not interacting with the apps directly. Nonetheless, the comparison of these two subscores across different categories give the users a contrasting picture of the current practice for data collection in context-specific lens.

3.5 Compare and Contrast Privacy Fingerprint and Intrusiveness Score

The Privacy Fingerprint and Intrusiveness Score are complementary metrics that help to assess and compare the intrusiveness among apps. They are built upon the same test data, so they reflect the same intrusiveness. However, due to the difference in presentation, the two metrics aid the privacy assessment process in different ways. The Privacy Fingerprint displays more details about the data access behavior of an app through the fingerprint visual, whereas the Intrusiveness Score captures the intrusiveness of that access behavior in a single number. Thus, for smartphone users wishing to make quick judgments about app choices, the Intrusiveness Score can help them arrive at a fast decision about which app to install, without needing to understand the types of data accesses that are involved. For users who are more curious and selective about which types of data are accessed by apps and when, the Privacy Fingerprint offers an explanatory aid to the Intrusiveness Score, as well as details about the app's data accesses which users can use to arrive at their own decisions about intrusiveness.

4. STUDY RESULTS

We apply our Intrusiveness Formula to the 33 apps tested in our experimental study and report their Intrusiveness Scores and Subscores in this section. In Table 2, we report the average Intrusiveness Scores for each of 4 app categories and provide a relative ranking based on those scores in order to gain a broad sense of intrusiveness across app categories. The greater proportion of access activity occurring during the user's idle periods causes Messaging and Social apps to have higher Intrusiveness Scores than others. We also delve deeper into the individual Intrusiveness Scores and Subscores of apps within the Games and Messaging categories and show how an app's level of idle access activity can affect the relative ranking of apps within each category.

4.1 Intrusiveness of Games Apps

Games apps exhibited great variance in overall Intrusiveness Scores, from scores as low as 1.00 to scores as high as 18.51. This great spread in scores reflects the differences in data access patterns of apps, as well as varying levels of idle access activity.

Table 3: Ranking and Intrusiveness Score of Game Apps

#	Apps	Score	#	Apps	Score
1	Angry Birds	18.51	6	Fruit Ninja	8.23
2	Scramble	12.18	7	UNO®	6.67
3	Fishing Star	10.61	8	Cut the Rope Free	6.52
4	Bow Man	10.08	9	Flow	5.45
5	Paper Toss	9.81	10	Temple Run	1.00

Table 4: Intrusiveness Subscore Breakdown of Game Apps

Game App	Active Score	Idle Score	Idle(%)
Angry Birds	8.05	20.05	87.2
Scramble	7.97	19.00	38.14
Fishing Star	7.36	19.00	27.87
Bow Man	8.09	19.29	17.72
Paper Toss	7.85	19.00	17.54
UNO®	6.01	17.00	6
Cut the Rope Free	6.00	18.00	4.37
Fruit Ninja	8.05	20.00	1.54
Flow	5.45	0.00	0
Temple Run	1.00	0.00	0

From Angry Birds' Privacy Fingerprint, we see (with reference to Table 1) that Angry Birds accesses wifi (15,16), imsi (13, 37), device id (11, 35), and gps (9,10, 33, 34). When the user is active, most of the accesses are devoted to static data whose values are fixed over time, such as imsi and device id. When the user is idle, however, there is a noticeable shift in data access patterns, with gps location taking up most of the requests. While most Game apps made sensitive data requests during the user's idle period, most of these accesses only account for a small percentage of the app's total accesses (7 out of 10 apps exhibited idle accesses at most 20% of the time). However, Angry Birds was the notable exception, with 87.2% of all its data accesses being made while the user wasn't even playing the game. This prominent idle access activity, coupled with the app's breadth of sensitive data accesses, landed Angry Birds as the most intrusive app in the Games category, a pronounced 6 points ahead of the next app on the list. Note that in active access activity, Bow Man's data accesses have greater context intrusiveness than Angry Birds' (See Figure 3 for their Privacy Fingerprints). Specifically, in addition to accessing gps location (like Angry Birds), Bow Man also accesses the user's browsing information, which ranks higher on the context intrusiveness than any other data accesses made by Angry Birds.

4.2 Intrusiveness of Messaging Apps

Messaging apps rank the highest in categorical intrusiveness due to the variety of sensitive data requested (sms, photo, audio, contact, email, gps location, device id, imsi, msisdn, wifi) and its prominent access activity during the user's idle period. For half of the Messaging apps tested, idle request activity contributed to at least 40% of all data accesses, with Kakao Talk (see Figure 4) dominating the category with 83.46%. Messaging apps also exhibit the smallest distribution in Intrusiveness Score, suggesting that messaging apps exhibit similar types of data access patterns. Google Voice and Facebook Messenger, products of two of biggest and wealthiest content providers in the world, rank the lowest on the intrusiveness scale. This low score results from the limited types of sensitive data they collect, as well as the low contribution of idle access activity (< 5% of all sensitive data accesses). Their conservative access behaviors suggest that data is only ac-

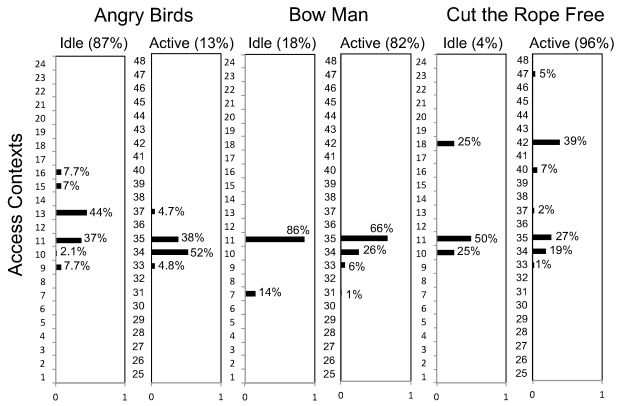

Figure 3: Privacy Fingerprints for Game Apps

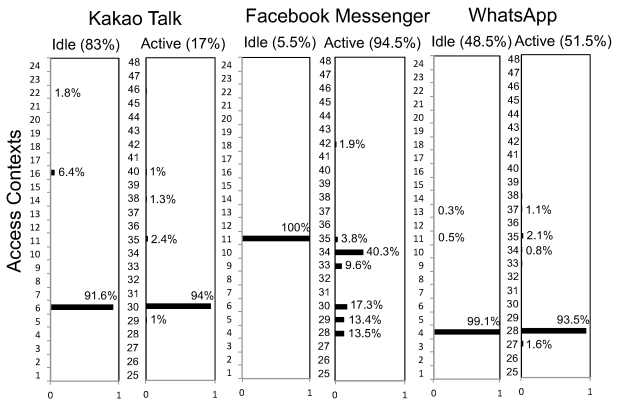

Figure 4: Privacy Fingerprints for Messaging Apps

Table 5: Ranking of Messaging Apps by Intrusiveness Subscores

#	Active	Idle	#	Active	Idle
1	Skype	Skype	5	Kakao Talk	Kakao Talk
2	Go SMS Pro	WhatsApp	6	Pinger	Handcent
3	Google Voice	Go SMS Pro	7	Facebook Messenger	Facebook Messenger
4	Whats App	Pinger	8	Handcent	Google Voice

cessed when necessary for app functionality. For example, Google Voice only accesses sensitive data (contact info, msisdn) while the user is active (and not idle).

5. DISCUSSION

One major contribution of the intrusiveness study is to demonstrate that apps not only vary greatly in the types of sensitive data they access, but also in the usage conditions under which they access that data, an often overlooked component in proposed transparency tools. These disparities account for differences in the Intrusiveness Scores of smartphone apps, both *across* and *within* app categories. Indeed, each app truly does have a singular fingerprint that uniquely quantifies its data access behavior. These differences in access behavior, and their subsequent impacts on relative intrusiveness, highlight the need for a method that allows users to assess apps that provide similar functionalities but exhibit highly variable sensitive access activity, so that they can easily and accurately distinguish more intrusive apps from less intrusive ones when choosing which apps to install.

Reputation and popularity vs. intrusiveness

One notable finding of the intrusiveness study is that popularity and reputation are not reliable indicators for reduced intrusiveness. All of the apps chosen for testing ranked in the top 20 apps for their respective categories. Each had at least 1,000,000 installations and ratings of at least 4 stars given by least 10,000 users, most with more than 100,000. Yet, the majority of them exhibited a significant level of idle access activity, many of them being for highly sensitive data (contact info and sms). Company reputation also does not guarantee lower privacy risk. Facebook application was found to be the second most intrusive app out of all Social apps tested. Thus, users should not automatically attach good privacy practices to the high reputability of app companies.

Table 6: Intrusiveness Subscore Breakdown of Messaging Apps

Messaging App	Active Score	Idle Score	Idle (%)
Kakao Talk	10.79	21.51	83.46
Pinger	10.40	21.95	70.91
Go SMS Pro	10.90	21.96	48.46
WhatsApp	10.86	21.97	40.1
Handcent	9.10	20.05	39.33
Skype	10.91	21.99	37.61
Facebook Messenger	9.35	19.00	5.45
Google Voice	10.90	0.00	0

Context-specific activities and intrusiveness

The level of idle access activity causes significant changes in relative intrusiveness ranking. This observation is exemplified by the top ranking app for intrusiveness in the Messaging category: Kakao Talk. Based on data access patterns alone, the app was ranked 5th out of 8 apps for intrusiveness. However, it rose to number one after the level of idle access activity was factored into the intrusiveness score, mostly due to the fact that the majority (83.46%) of its accesses were performed while the user was idle. Even though Kakao Talk accessed fewer highly identifiable data types than other Messaging apps (e.g., Facebook, Google Voice, Skype, WhatsApp), its significantly higher level of idle access activity made it more intrusive than any other app tested in the Messaging category.

This suggests that the level of idle activity is sometimes more impactful on relative intrusiveness than the types of data that are accessed. This is especially true for apps that access similar types of data, such as apps within the same category. Thus, the level of idle activity is a highly relevant factor for assessing intrusiveness, because users tend to compare apps within the same category when choosing which apps to install. This is an important finding, because current transparency mechanisms (Android Permissions Model, TaintDroid [3]) only focus on identifying the types of data that are accessed by apps, and not the usage contexts under which they are accessed.

6. RELATED WORK

Limitations of Existing Transparency Models

Various efforts have been made by mobile platforms to mitigate privacy issues. For example, Android's permission framework requires developers to specify each piece of sensitive data required to run their application, asking smartphone users for all the necessary permissions upon installation. Furthermore, app devel-

opers can specify their privacy policy[4] to gain more trust from the users. However, interviews with Android users demonstrate that these documents are largely ignored and unhelpful at conveying privacy risk [10], most likely due to lengthy descriptions and technical jargon. Interviews conducted by King [11] show that most smartphone users exhibit an over-reliance on existing assurance structures, erroneously believing that the reputability of a platform provider guarantees a thorough screening process for filtering out rogue apps.

Many transparency tools have also been built by researchers to monitor sensitive information collected by smartphone apps. While these frameworks are useful for exposing the types of data collected by smartphone apps, their use is currently limited to the *identification* of sensitive [3, 7] and possibly malicious transactions [5], leaving out the contexts under which they are obtained and thus overlooking an important component in evaluating privacy risk [1].

Measuring Privacy and Privacy in Context

Modeling people's privacy concerns for using smartphone apps is not easy. Mancini et al. [13] discussed the challenges of modeling mobile privacy, and investigated the notion of "context" in relation to people's privacy experience. The Intrusiveness Score partly captures the "context-deviation behavior" of an app that violates the principle of "respect for context" [17].

Lin et al. [12] employs user expectation to model privacy, thus providing smartphone users with a more instinctual metric for assessing privacy risk. The study uses crowdsourced user surveys to gauge the unexpectedness of certain data accesses, juxtaposing perceived app functionality with the actual permissions that are required to install the app. While crowdsourcing makes their approach more scalable, crowd opinions are not necessarily reliable indicators of truly unwanted access behavior. As many studies have shown [11, 2], most smartphone users cannot accurately grasp the privacy implications of sensitive app accesses based on permissions alone. We introduce the notion of "privacy in context" [15, 17] into our analysis of apps' behaviors for intrusiveness and quantify the sense of intrusiveness without asking the user privacy questions directly (or indirectly) that could bias the results.

7. CONCLUSION AND FUTURE WORK

We have developed a privacy assessment framework that focuses on quantifying intrusiveness using two newly developed metrics: 1) the Privacy Fingerprint visual that characterizes the data access breakdown of an app and 2) an Intrusiveness Score that calculates its intrusiveness based on various privacy-relevant intrusiveness factors. The results from our study of 33 popular Android apps demonstrate that the Intrusiveness Score is especially useful for comparing the privacy risk of apps exhibiting similar types of data accesses, a limitation of current studies. Furthermore, our findings show that idle access activity, largely overlooked by current transparency structures, are prevalent yet likely unexpected practices among popular apps, drastically affecting their relative intrusiveness scores within a given app category and impacting a user's decision-making process during app installation. Though we believe our new privacy assessment framework provides a promising start for quantifying privacy risk, the Intrusiveness Formula needs more refinement. Specifically, the datatype identifiability weights and sequence of intrusiveness factors used to sort the access context list by increasing intrusiveness need to be validated through more rigorous empirical testing. Furthermore, user studies should be conducted to verify the distribution of access contexts, as well as the percentages of active and idle access activity, so that they account for possible variations in user-specific behavior.

[4] `http://support.google.com/googleplay/android-developer/bin/answer.py?hl=en&answer=2519872`

8. REFERENCES

[1] D. Boyd. Making sense of privacy and publicity, Mar. 2010.

[2] E. Chin, A. P. Felt, V. Sekar, and D. Wagner. Measuring user confidence in smartphone security and privacy. In *ACM SOUPS '12*, 2012.

[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. OSDI'10. USENIX Association, 2010.

[4] Federal Trade Commission. Protecting Consumer Privacy in an Era of Rapid Change, March 2012.

[5] A. P. Fuchs, A. Chaudhuri, and J. S. Foster. SCanDroid: Automated Security Certification of Android Applications. Technical Report CS-TR-4991, University of Maryland, 2009.

[6] R. Gavison. Privacy and the Limits of Law. *Yale LJ*, 1979.

[7] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall. These aren't the droids you're looking for: retrofitting android to protect data from imperious applications. CCS '11. ACM, 2011.

[8] K. Irwin and T. Yu. An identifiability-based access control model for privacy protection in open systems. In *Proceedings of the 2004 ACM workshop on Privacy in the electronic society*. ACM, Oct. 2004.

[9] M. M. Jan Lauren Boyles, Aaron Smith. Privacy and data management on mobile devices. Technical Report CS-2011-02, Pew Research Center, September 2012.

[10] P. G. Kelley, S. Consolvo, L. F. Cranor, J. Jung, N. Sadeh, and D. Wetheral. A conundrum of permissions: Installng applications on an Android smartphone. In *Proceedings of the Workshop on Usable Security (USEC)*, Feb. 2012.

[11] J. King. "How come I'm Allowing Strangers to Go Through My Phone?": Smart Phones and Privacy Expectations. In *ACM SOUPS '12*, 2012.

[12] J. Lin, S. Amini, J. I. Hong, N. Sadeh, J. Lindqvist, and J. Zhang. Expectation and purpose: Understanding users' mental models of mobile app privacy through crowdsourcing. In *ACM Ubicomp'12*.

[13] C. Mancini, K. Thomas, Y. Rogers, B. Price, L. Jedrzejczyk, A. Bandara, A. Joinson, and B. Nuseibeh. From spaces to places: emerging contexts in mobile privacy. *ACM Ubicomp '09*.

[14] E. McCallister, T. Grance, and K. Scarfone. Guide to Protecting the Confidentiality of Personally Identifiable Information (PII) -Recommendations of the National Institute of Standards and Technology. Technical report, NIST, April 2010.

[15] H. Nissenbaum. *Privacy in context: Technology, policy, and the integrity of social life*. Stanford Law Books, 2009.

[16] P. Schwartz and D. Solove. The PII Problem: Privacy and a New Concept of Personally Identifiable Information. *New York University Law Review*, Aug. 2011.

[17] White House. Consumer data privacy in a networked world: A framework for protecting privacy and promoting innovation in the global digital economy. February 2012.