Simon Koch*, Malte Wessels, Benjamin Altpeter, Madita Olvermann, and Martin Johns

Keeping Privacy Labels Honest

Developer conformity to self-declared data collection via Apple Privacy Labels

Abstract: At the end of 2020, Apple introduced privacy nutritional labels, requiring app developers to state what data is collected by their apps and for what purpose. In this paper, we take an in-depth look at the privacy labels and how they relate to actual transmitted data.
First, we give an exploratory statistically evaluation of 11074 distinct apps across 22 categories and their corresponding privacy label or lack thereof. Our dataset shows that only some apps provide privacy labels, and a small number self-declare that they do not collect any data. Additionally, our statistical methods showcase the differences of the privacy labels across application categories.
We then select a subset of 1687 apps across 22 categories from the German App Store to conduct a no-touch traffic collection study. We analyse the traffic against a set of 18 honey-data points and a list of known advertisement and tracking domains. At least 276 of these apps violate their privacy label by transmitting data without declaration, showing that the privacy labels' correctness was not validated during the app approval process. In addition, we evaluate the apps' adherence to the GDPR in respect of providing a privacy consent form, through collected screenshots, and identify numerous potential violations of the directive.

Keywords: Smartphones, iOS, Apple, GDPR, Privacy

DOI Editor to enter DOI
Received ..; revised ..; accepted ...

***Corresponding Author: Simon Koch:** Technische Universität Braunschweig, Institute for Application Security, E-mail: simon.koch@tu-braunschweig.de
Malte Wessels: Technische Universität Braunschweig, Institute for Application Security, E-mail: malte.wessels@tu-braunschweig.de
Benjamin Altpeter: Datenanfragen.de e. V., E-mail: benni@datenanfragen.de
Madita Olvermann: Technische Universität Braunschweig, Industrial/Organizational and Social Psychology, E-mail: madita.olvermann@tu-braunschweig.de
Martin Johns: Technische Universität Braunschweig, Institute for Application Security, E-mail: m.johns@tu-braunschweig.de

1 Introduction

Smartphones are ubiquitous [16], and ever more services rely on smartphone ownership, e.g., in the forms of banking apps or messaging application. Smartphones are carried everywhere and have become part of our day-to-day clothing. They carry data that provides deep insights into our private life, including our contacts, pictures, browsing behavior, and where we spend our time.

Contacts and contact interaction provide information about who is in our social network and possibly on the types of relationships between us [22, 43]. As most current smartphones include GPS and a myriad of other sensors, they observe and record where we go every day and for how long we stay [37, 47]. Finally, any tokens that are unique to a user can cross-identify a user across different data collectors. Combining identifying tokens with privacy-sensitive data presents a huge threat for the smartphone user's privacy.

Privacy is heavily contested. The EU introduced the GDPR law in 2016, and made it mandatory in 2018 [12]. This law requires that a user has to explicitly agree to any personal data collection, in the context of an app, that is not necessary to provide a service or that the service provider has no legitimate interest for. However, the required changes to applications are not always effected, and the industry keeps collecting our data regardless [1].

Apple positions privacy among the company's core values, so they 'design Apple products to protect [users'] privacy and give [them] control over [their] information' [5]. Part of their privacy protection mechanism is asking app developers to specify their data usage practices via 'Privacy Nutrition Labels' (short: privacy labels) [3, 8].

Privacy labels are a method of displaying how an application collects and uses data [32]. They have been shown to impact users' awareness of privacy in the context of IoT devices [27, 28].

In this paper, we cast a first light on the state of privacy labels. Specifically, we investigate (1) how privacy labels are used in practice, and (2) whether developers adhere to their self-declared privacy labels. To this end, we make the following contributions:

- We present a comprehensive exploratory statistical analysis of privacy labels for 11074 apps across 22 categories, including three case studies for the categories Games, Finance, and Social Networking.
- We conduct a no-touch traffic collection of 1687 apps across 22 categories, leveraging honey data, with two purposes:
  - to compare the privacy label declarations with the actually transmitted honey data, and
  - to crosscheck the observed traffic against a set of known tracker domains.
- Finally, we evaluate app compliance with the GDPR by checking whether a data-collecting app displays a GDPR or privacy dialogue prior to collection.

To enable these studies, we provide the following technical advancements:

- infrastructure to conduct large-scale iPhone traffic interception, and
- a system to automatically detect privacy-label violation via traffic analysis.

Our experiments uncovered apps in which a privacy label validation did not take place during the Apple app store approval process. These apps clearly violate their labels by transferring information that has not been declared in the respective label.Additionally, we detect differences in collection behavior across different categories in the privacy labels and apps.

The remainder of the paper is structured as follows: We first discuss the legal context of data collection (Section 2) and how Apple's privacy labels are structured (Section 3). Then, we explain how we collected and analyzed a large set of privacy labels (Section 4) and what information they contain (Section 5). After the privacy label analysis, we detail our traffic collection framework and collection process, as well as the app dataset that we used (Section 6). We then present an analysis of the traffic collected (Section 7) and discuss its implications (Section 8). These findings are contextualized by the limitations of our work (Section 9) and related work (Section 10). Finally, we summarize our key contributions and results, developing a perspective towards possible future work (Section 11).

2 GDPR Legislation

The General Data Protection Regulation (GDPR) is a European Union (EU) law that came into effect on the 25 May, 2018 [18]. It applies to the processing of personal data of persons in the EU and European Economic Area (EEA).

Personal data is defined as 'any information relating to an identified or identifiable natural person'. This definition also includes any unique identification number such as advertising IDs, location, or credit card numbers [18], and has been interpreted accordingly in previous work [23, 38].

To be able to legally process personal data, the controller, i.e., the party deciding on the processing, must have a legal basis. Such a basis is also necessary if a third-party, called a processor, is doing the processing on behalf of the controller, e.g., in the case of tracking companies.

A controller or processor can process personal data if they are legally or contractually required to (e.g., to provide the app service), perform a duty in the public interest, have a legitimate interest, or if processing is necessary to protect the vital interests of the data subject. If none of these legal bases apply (e.g., for advertisements [10]), explicit consent has to be given by the affected user.

Concerning consent, GDPR Art. 7(2) [7] states that *if the data subject's consent is given in the context of a written declaration which also concerns other matters, the request for consent shall be presented in a manner which is clearly distinguishable from the other matters, in an intelligible and easily accessible form, using clear and plain language. Any part of such a declaration which constitutes an infringement of this Regulation shall not be binding.* This implies that the simple act of starting an app, or granting it permissions via the operating system, cannot constitute consent. Furthermore, the consent has to be explicit and a user has to have a meaningful choice, i.e., be able to opt-out without consequences.

3 Apple Privacy Labels

On 14 December 2020, Apple announced privacy labels on their App Store [4] and introduced a privacy information section for each app. This is meant to give users key information on what data is collected and how it is used, i.e., whether for tracking, and whether it is linked to, or not linked to, the user [3, 8]. It is important to note that

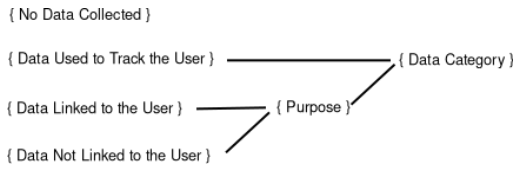


Fig. 1. The structure of a privacy label. It starts leftmost with the privacy type. Followed by the purpose of data collection. Then rightmost the collected data categories. A connection indicates that the left element contains a list of the right element.

privacy labels are self-declared and, thus, inherit the trust that users place in the developer of the app itself. As far as we are aware, there is no entity fact-checking the labels.

In this section, we go into detail on how Apple structured the privacy labels and what information we can gain by studying them. The presented information is based on the developer documentation by Apple [3].

3.1 Intended Contained Information

Apple requires all apps to have privacy labels if uploaded or updated after their introduction. App developers thus have to identify any data collected from the user. Apple deems data *collected* when it has left the phone and is stored longer than minimally required to answer its immediate use.

The information is not limited to the data collected by the app itself, but is supposed also to identify the data collected by third-party partners or SDKs. Apple also stresses that an app's privacy practices should follow all applicable laws and that developers are responsible for keeping details accurate and up to date [3].

We give a visual overview of the structure of a privacy label in Fig. 1.

3.2 Structure of the Privacy Labels

The main categories of privacy labels are the privacy types. They explain how the data is collected and processed.

3.2.1 Privacy Types

There are four different privacy types. The same privacy label can contain different privacy types except for *No Data Collected*, which mutually excludes any other label and does not provide further details.

No Data Collected: This privacy type is not further detailed, and simply states that the app does *not collect* any details.

Data Used to Track the User: This privacy type covers data collected for *tracking*. Tracking is defined as linking collected data with third-party data for targeted advertising or for measuring advertising outcomes. Additionally, the tracking label also includes data collected and then shared with data brokers. This privacy type contains a list of data categories collected.

Data not Linked to the User: This privacy type covers collected data that is *not linked* to the user. Apple explicitly states that data collected from an app is often linked to a user unless anonymization, such as a stripping process of user IDs, is put in place. They also stress that any action that either links the user's identity back to the data, or that combines the collected data in a form that allows linking back to the user's identity, excludes collected data from this category. This privacy type contains a list of purposes, each containing a list of the collected data categories.

Data Linked to the User: This privacy type covers collected data *linked* to the user, i.e., the collection does not fit the definition of data not linked to the user. It contains a list of purposes each containing a list of the collected data categories.

3.2.2 Purposes

Both of the user-*linked* and -*not linked* privacy types list the collected data by purpose. There are six different purposes:

Third-Party Advertising: Data used to display third-party ads in the application, or data shared with third-party advertisers who display third-party ads.

Developer's Advertising: Data used to display first-party ads, used for marketing directly to the user, or data shared with vendors directly displaying the developer's advertisements.

Analytics: Data used to evaluate user behaviour and characteristics. Examples are A/B testing or audience analytics.

Product Personalization: Data collected for this purpose is used to personalize the product for a user. Examples are recommendations or suggestions.

App Functionality: Data collected for this purpose is required for the app's functionality. Examples are authentication or customer support.

Other Purposes: Any purpose not covered by the other purposes.

3.2.3 Data Categories

Apple defines multiple data categories, each containing a list of corresponding *collected* data type names, such as `Device ID` or `Email`. The data categories are `Contact Info`, `Health & Fitness`, `Financial Info`, `Location`, `Sensitive Info`, `Contacts`, `User Content`, `Browsing History`, `Search History`, `Identifiers`, `Purchases`, `Usage Data`, `Diagnostics`, and `Other Data`.

We detail only the categories that contain unexpected data types or whose names are not self-explanatory:

Sensitive Info only contains the data type `Sensitive Info` itself, which sounds diffuse. Apple defines it as any data that relates to ethnicity, sexual orientation, pregnancy/childbirth, disability, religious or philosophical beliefs, trade union membership, political information, or biometric data.

Health & Fitness contains two data types: `Health` and `Fitness`. `Health` relates to both health and medical data, including data from the clinical health records API, HealthKit API, MovementDisorder API, or further health-related human subject research or otherwise user-provided health or medical data. `Fitness` includes data from the Motion and Fitness API.

Both categories may include further data from other sources as well. Both categories are broad, and their definitions do not comprehensively cover what they contain.

Browsing History contains information about content the user has viewed that is not part of the app, such as websites. iOS does not provide any APIs to read the browsing history of Safari.

3.2.4 Optional Data Disclosure

Apple also defines data collection that is optional to disclose, splitting it into three distinct groups.

General Data: A developer may choose not to disclose general data that

- is not used for tracking (i.e., not linked with third-party data for advertising measurement purposes, nor shared with a data broker), and
- is only collected infrequently and is not part of the app's primary functionality, and
- is provided by the user via the app interface, and it is clear to the user what data is collected.

If collected data meets all of these criteria, the developer has the option of disclosure via the privacy labels. However, Apple does stress that "data collected on an ongoing basis after initial request for permission must be disclosed".

The two additional exceptions are Regulated Financial Data and Health Research Data, which, under special circumstances (e.g., if legally required), do not require disclosure.

4 Privacy Labels Analysis

In this section, we detail our approach to retrieving, preparing, and analysing our data privacy label dataset of 17482 apps.

4.1 Collection of Privacy Labels

First of all, we need to create a large collection of apps to analyze. We use the 3u web API to request a list of all app IDs across different categories, ordered by absolute rank [2]. The categories provided by 3u are: books, business, education, entertainment, finance, food and drink, games, health, lifestyle, medical, music, navigation, news, photo and video, productivity, references, shopping, social network, sports, travel, utilities, and weather. Those categories and rankings are curated by 3u, and are not necessarily identical with those in the Apple App Store. 3u lists the first 1000 apps in each category. After accounting for redundancies across categories, we have a list of 17482 different app IDs across 22 categories.

Using the list of app IDs, we then access an Apple-provided web API to request the privacy label of each app in JSON format. (The labels analysed in This study were collected in November 2021.)

4.2 Clean-Up

Overall, there were 11812 apps without any privacy label, due either to the app not being accessible in the German App Store (6408), or through not have been assigned a label yet (5404). On average, each category is missing 670.23 privacy labels with a standard deviation of 111.35. Table 2 summarizes label accessibility.

The category with the most labels absent is *References* with 422 labels missing. Among apps not available in the German App Store, *Shopping* is the largest category with 553 inaccessible apps. The category with the least number of labels or apps missing is *Games* (422

Table 1. Distributions of privacy types, as well as the least and most popular purpose and data types.

(a) Prevalence, average, and standard deviation of the different privacy types split by categories.

Privacy Type	Total	Avg.	Std. Dev.	Most	Least
No Data Collected	823	48.27	17.80	Business (155)	Games (5)
Data Not Linked to the User	2143	127.82	16.35	Photo and Video (155)	Business (90)
Data Linked to the User	1746	105.27	23.38	Games (163)	Weather (66)
Data Used to Track the User	1098	65.36	36.39	Games (189)	Business (23)

(b) Prevalence, average, and standard deviation of the most and least popular purpose by privacy type, aggregated across categories

Privacy Type	Most	Avg	Std Dev	Least	Avg	Std Dev
Date Not Linked to the User	App Functionality (1613)	96.95	12.01	Other Purposes (220)	13.00	5.59
Data Linked to the User	App Functionality (1604)	97.36	24.87	Other Purposes (255)	15.95	7.78

(c) Prevalence, average, and standard deviation of the most and least popular data types split by privacy type aggregated across purposes and categories.

Privacy Type	Most	Avg	Std Dev	Least	Avg	Std Dev
Data Not Linked to The User	Crash Data (1543)	92.09	15.55	Other Financial Info (2)	0.09	0.29
Data Linked to the User	User ID (1184)	72.05	23.72	Credit Info (24)	1.68	3.48
Data Used to Track the User	Device ID (795)	47.00	33.25	Health (0)	0	0

and 106). After removing the apps without labels, we are left with a data set containing 5670 distinct apps, with, on average, 329.77 apps per category and a standard deviation of 111.35. The category with the fewest overall accessible privacy labels is *Food and Drink*, with 211 labels.

We ensure comparability of the different categories by choosing 211 apps with labels from every category, according to the rank provided by 3u, to ensure that each category subset has the same number of privacy labels.

4.3 Analysis Method

Table 2. The distribution of all apps provided by 3u concerning their privacy label accessibility. Note that the averages are calculated over the whole category (i.e., 1000 apps) whereas the absolute numbers are given for unique apps.

Category	Apps	Apps %	Avg	Std. Dev.
All Apps	17482	100.00%	1000	0
has Label	5670	32.43%	329.77	111.35
missing	11812	67.57%	670.23	111.35
no Label	5404	30.91%	317.18	73.17
no App	6408	36.65%	353.05	84.45

To our knowledge, there are no prior investigations of Apple's privacy labels, thus, our analysis cannot serve any specific prior hypothesis. There are thus a large number of possible distribution hypothesis tests that

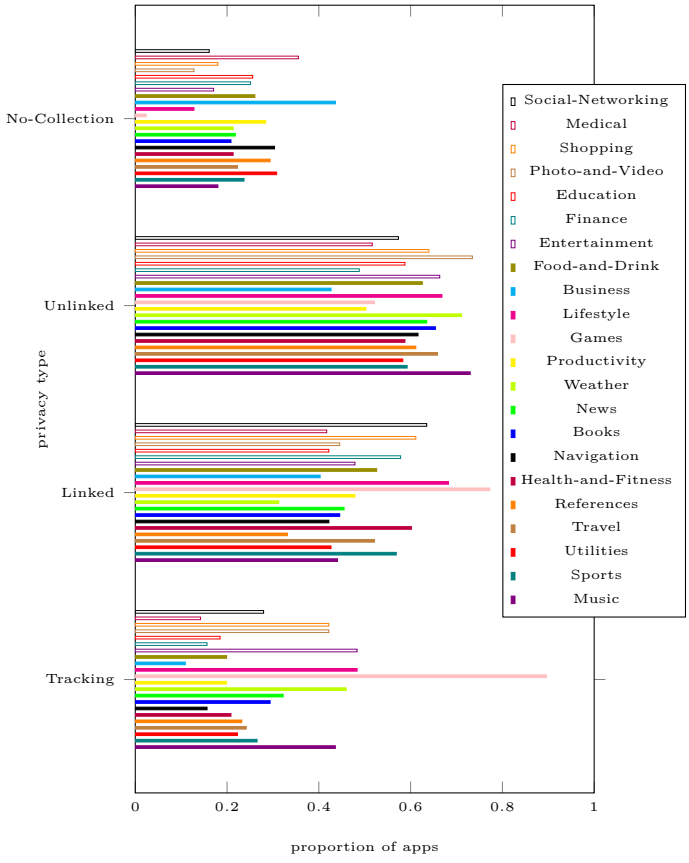


Fig. 2. A visual representation of the usage of the different privacy types across the app categories. Note that No-Collection is mutually exclusive with every other category.

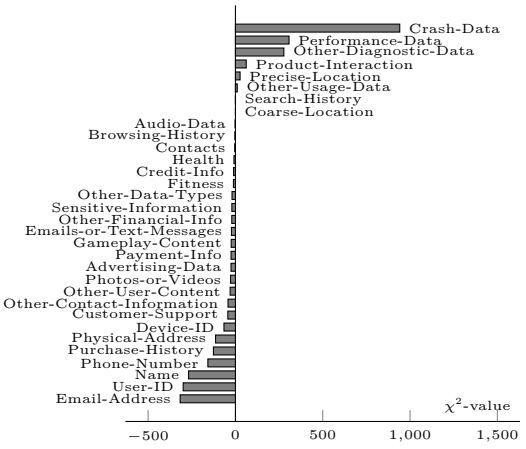
could be applied, and applying them would be inherently prone to an inflation of the false positive rates due to the multiple testing problem [31]. To avoid this methodical flaw, we forego comparative statistics in favour of purely descriptive methods.

In order to present a first impression of privacy labels across (1) app categories, (2) purposes and (3) privacy types, we present the average occurrence, standard deviation, and prevalence. First of all, we present a visual representation for the privacy types across different categories in Fig. 2 and the raw results in Table 1a.

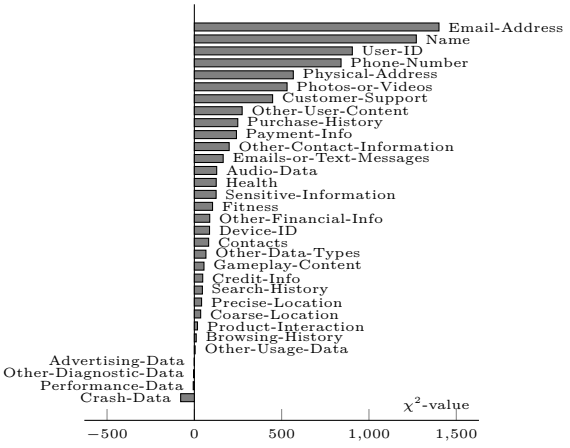
Secondly, prevalence for the different purposes by privacy type aggregated across categories can be found in Table 1b. Table 1b lists the averages and standard deviations for the most and least popular purpose of the privacy types, 'Data Collected Linked to the User' and 'Data Collected not Linked to the User'. The privacy types 'Data Used to Track the User' and 'No Data Collected' do not contain any purposes, and are thus left out of this analysis.

Lastly, we analyze data types aggregated across categories and purposes, but split by privacy type. Table 1c lists the averages and standard deviations for the most and least popular data types for each privacy type. The privacy type, 'No Data Collected' does not contain any data types and is thus omitted from this analysis.

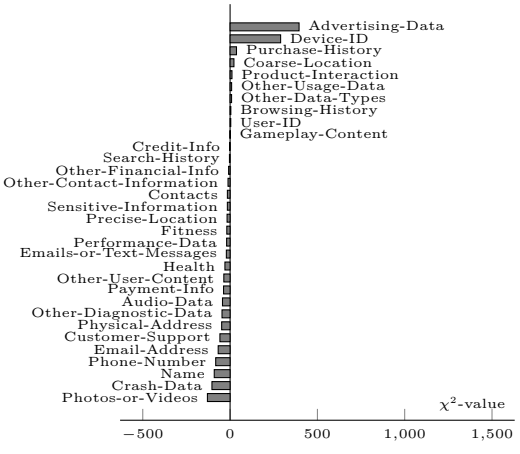
Based on this first descriptive insight into the available data set, we use keyness plots in order to identify differences and similarities between categories [44]. Keyness plots originate in linguistics, where they are used to compare word frequencies between sample and reference documents. χ^2 values are commonly used as an indicator of how much the frequency of a word differs between two compared documents. We adapt this method to compare privacy labels and their attributes across app categories to examine the extent to which a data type, purpose, or privacy type is more prevalent in one category than in the others. Specifically, we generate keyness plots by singling out an app category of interest, pooling the remaining categories, and then calculating chi^2 assuming the average distribution across all categories as the reference distribution. If the count for the singled-out category value is larger than the expected average under the assumed distribution, χ^2 is plotted positively, i.e., signaling keyness in favor of the category; if it is smaller, then the value is plotted negatively, i.e., signaling keyness against the singled-out category. The calculation is demonstrated, via an example, in the Appendix. Fig. 3 provides three keyness plots comparing the collected data categories for the different privacy types.



(a) Differences in collecting data types *not linked to the user.*



(b) Differences in collecting data types *linked* to the user.



(c) Differences in collecting data types for *tracking.*

Fig. 3. Keyness plots showing differences in the collection frequency of different data categories across different privacy types. The positive values indicate a higher prevalence (higher keyness), whereas negative values show a lower prevalence (lower keyness).

5 Data Collection Patterns

In this section, we provide a first insight into the various data collection patterns according to the privacy type and data type. To begin with, we address missing labels, uncovering inconsistencies in the data collection declared by the developer. Next, we discuss differences in the distribution of privacy labels in our analysis. Then, we discuss our three case-study categories (namely games, finance, and social networking) to more closely assess differences between app categories and their data collection practices according to the privacy type and data types. Finally, we discuss the implications of a privacy label in the context of the GDPR and close with a summary of *lessons learnt* .

5.1 Missing Labels

The overall number of apps missing privacy labels is not negligible, at 48.87% of all apps available on the German App Store. This shows that the overall adoption of the privacy label is a still ongoing process.

Apple's current policy is to require a privacy label upon the next update after privacy labels were introduced. This policy needs further refinement to ensure that all apps do eventually receive a privacy label, even if updates are infrequent, as we are now nearly a year into the adoption of privacy labels in the App Store. One easy option would be to require developers to update their apps with a privacy label within the next quarter, and to be excluded from the App Store otherwise.

5.2 Inconsistencies

In our analysis of the collected privacy labels, we encountered several problematic inconsistencies:

(1) Developer can inconsistently claim to collect non-anonymizable data points as 'not linked to the user': Privacy labels allow the assignment of any data type to any privacy type, e.g., a developer can declare that it will collect a `User ID` *not linked* to the user.

In our dataset, 713 apps claim to collect the `Device ID`, 201 the `Email`, 344 the `User ID`, and 91 the `Phone Number` without linking them to the user. We consider this to be impossible, as every one of those categories is synonymous with user identity.

(2) Developer can inconsistently claim to collect app personalization data not linked to the user: Another inconsistency permitted by the privacy labels is to collect data *not linked* to the user for product personalization (Fig. 4). It does not seem possible or plausible to collect data in such a fashion as data must not be linked back to the user's identity, which is at odds with personalizing.

The observed inconsistencies show that Apple is not checking every app for contradictory declarations, and it may well be that Apple is not checking at all. We would expect that Apple performs at least minor sanity checks on the labels catching such contradictory declarations. Apple should either improve its documentation to explain how such declarations work, or ensure that implausible declarations are impossible.

5.3 Data Collection by Type and Purpose

Overall, app categories are consistent in what types of data they collect. Data *linked* and *not linked* to the user are both popular with 105.27 and 127.82 apps, on average, across the categories. *No data collection* is unpopular, with an average of only 48.27 apps per category. *Tracking* is the most inconsistent privacy type, with an average of 65.36 apps and a standard deviation of 36.39.

The popularity of the data types collected differs between the different privacy types, i.e., whether the data is collected for *tracking*, *linked* to the user, or *not linked* to the user. For *tracking* collecting, the `Device ID`, `User ID`, `Advertising Data`, or `Product Interaction` types are the most popular. All of these data points are relevant for optimizing advertising or to evaluate its efficacy, so consequently collection is to be expected for tracking. However, a non-negligible number of apps (239) declared that they collect `Crash Data` for *tracking*. This is counterintuitive, and warrants further investigation in future work. We suspect that this is a declaration mistake. The most-collected data types *linked* to the user the `Email`, `Name`, `User ID`, `Device ID`, and `Product Interaction`. For data *not linked* to the user, `Crash Data`, `Performance Data`, `Product Interaction`, `Device ID`, and `Other Diagnostic Data` are the most frequently collected. Fig. 14 in the Appendix contains the corresponding plots.

The different data types collected—either *linked*, *not linked* to the user, or for *tracking*—paint a picture in which data for debugging and improving an application is *not linked* to the user, whereas data for personalization or to target advertising is *linked*, and indirect advertising data is used for *tracking*.

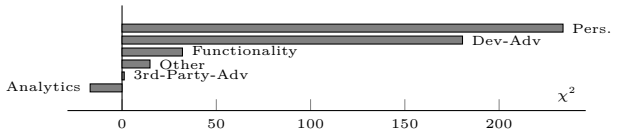


Fig. 4. Keyness plot comparing the frequency of purposes *linked* (positive) and *not linked* (negative) across all apps.

App functionality is about equally common for both *linked* and *not linked* data collection, which aligns with its explanation including data that is *linked* to the user (e.g., for authentication) or is *not linked* to the user (e.g., for minimizing app crashes). Analytics data is more frequently *not linked* the user, which makes sense, as evaluating user behavior or understanding the effectiveness of app features can be evaluated with anonymized data. We provide corresponding plots in Fig. 13 in Appendix.

Our initial analysis is supported by plotting the keyness of the different data types (Figures 3a–3c), clearly showing that `Crash Data` as well as `Performance Data` and `Diagnostic Data` have high keyness for *not linked* to the user, `Advertising Data` and `Device ID` have high keyness for *tracking*, and `Email` as well as `Name` have high keyness *linked* to the user. Plotting the keyness of purposes user-*linked* and *not linked* (Fig. 4) further strengthens our analysis, as `Personalization` and `Developer Advertising` have high keyness for collected user-*linked* data. `Analytics` has a high keyness for data collection *not linked* to the user, but the keyness for purposes *linked* to the user is markedly larger, most likely due to the unambiguous nature of the corresponding data collection, whereas `Analytics` can reasonably contain both user-*linked* and *not linked* data, and thus has a noticeably lower keyness.

Finally, analyzing the number of apps collecting data types corresponding to their purpose, split into *linked* and *not linked* groups, shows that the most frequently collected data types *not linked* to the user, for `Analytics` or `App Functionality`, are `Crash Data`, `Product Interaction`, and `Performance Data`. The most frequently collected data types *linked* to the user for `App Functionality` are data points such as `Name`, `Email` and `User ID`, which would be required for user customisation. A complete plot showing collection of all data categories for individual purposes is given in the Appendix, Fig. 15.

5.4 Case Studies

Due to the high numbers of possible analyses and comparisons between categories, we chose three exemplary categories to present in-depth: Games, Finance and Social Networking. Gaming apps are often free of charge and rely on advertisements to cover their expenses, which might be observable in privacy label patterns. Finance apps include sensitive data and are therefore a privacy risk in principle. Lastly, social networks are known for large-scale personal data collection, so Social Networking apps are of interest for a possibly higher level of private data collection and advertising related tracking. These three categories offer highly interesting cases to analyze and assess common stereotypes. Please note that the subsequent keyness plots depict only one app category with its privacy types and data types collected, in comparison to all other categories, and do not indicate differences between the categories.

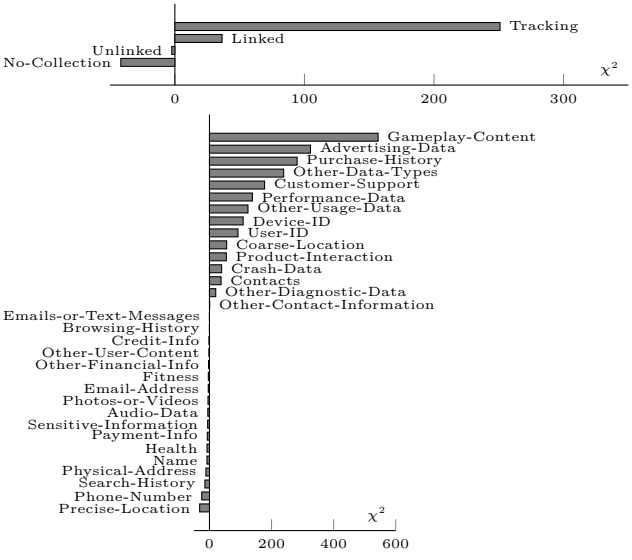


Fig. 5. Keyness plots comparing apps in the Games category with the remaining apps concerning privacy types and data types.

Let us start with a closer look into the raw counts for our app categories. One category noticeably differs from the others: Games. Gaming apps most frequently collect data for *tracking* (189) and data *linked* to the user (163). As visualized in Fig. 5, the high values of χ^2 for the data types 'Linked' and 'Tracking' indicate overproportional tracking and linkage of data compared to all other categories. Only the collected data *not linked* to the user by apps in the Games category is little different from the average; even slightly

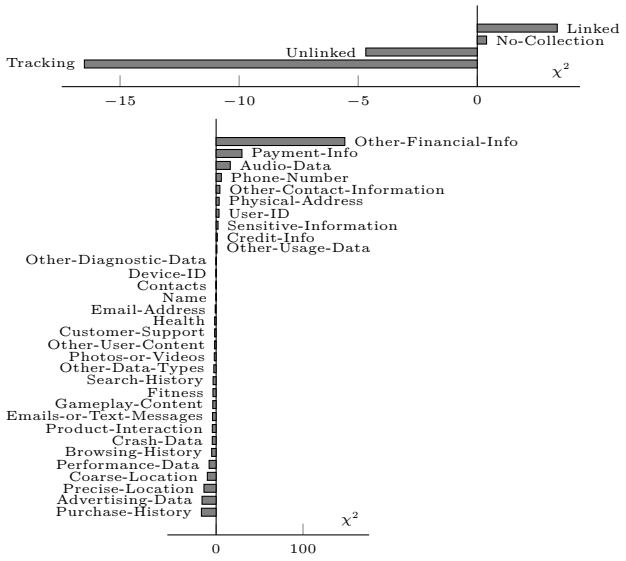


Fig. 6. Keyness plots comparing Finance apps with the remaining apps, concerning privacy types and data types. Note the different scales on the x axis.

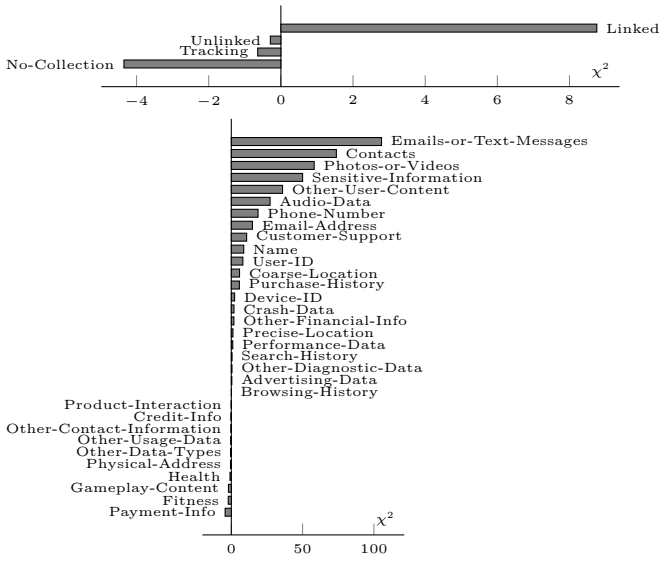


Fig. 7. Keyness plots comparing Social Networking apps with the remaining apps, concerning privacy types and data types. Note the different scales on the x axis.

lower. This paints a picture of the Games category being the most active in collecting data related to the user, at a level markedly different from the other categories. But gaming apps also do collect data types according to their purpose, such as `Gameplay Content`. Beyond that, advertising-related data has a noticeably high keyness (e.g. `Purchase History`, `Advertising Data` and `Product Interaction`; see Fig. 5). This demonstrates a high prevalence of advertising-related data collection in comparison with the other app categories, whereas the remaining types of data collected are less different from those in the other categories.

In contrast to the prior plots of gaming apps, Finance apps reveal less *tracking*, whereas data *linked* to the user is collected more than compared to all other categories (Fig. 6). It has to be noted, however, that χ^2 and, therefore, the indicator for the extent of the difference, is only a fraction of the keyness values for games (Fig. 5). This further underlines the extent of deviating collection patterns by gaming apps. Financial apps necessarily collect substantially more `Other Financial Info` to fulfil their main purpose, whereas all other types of data collection are not that much different from the other two categories.

Lastly, the Social Networking category differs mainly in the number of apps collecting *linked* data (Fig. 7). Social Networking apps do collect sensitive data for the intended purpose to connect with other people and to enlarge personal networks. We notice a large difference in personal data types such as `Videos And Photos`, `Sensitive Information`, `Audio Data`, `Phone Number`, `Contacts` and `Email`. Surprisingly, `Advertising Data` is not considerably different to all other categories, which contradicts stereotypes of social media profiteering through advertising.

5.5 Declared Data and the GDPR

We now briefly assess what data items can be declared in a privacy label, towards discerning what privacy label declarations should be followed by obtaining user consent before the app can actually start collecting the data.

Prima facie, any data *not linked* to the person, assuming sufficient anonymization, should not be affected by the GDPR. However, data points such as `Device ID`, or `User ID` can hardly be anonymized sufficiently to be considered unrelated to an identifiable natural person. Consequently, regardless of how the collection of those data points is declared, we expect them to fall under GDPR protection. The remaining two collection types, data *linked* to the user and data used for *tracking*, imply by their names that the data is related to an identified natural person and should therefore fall under GDPR protection.

This assessment is fairly superficial, however, as the GDPR explicitly permits processing of legally required data. An assessment taking this into consideration would have to be done on case-by-case and country-by-country basis, exceeding the scope and focus of this work.

5.6 Lessons Learnt

Overall, apps collect data types for the purposes and privacy types that one would expect. Only few apps *do not collect* any data, and games are especially active in data collection and *tracking*. There is some inclination to collect data in a way that preserves the user's privacy, i.e., *not to link* data back to the user if not required (e.g., in analytics). However, this inclination is small and opposed by apps collecting data types clearly *linked* to the user while deeming them *not linked*.

We found inconsistencies in the labels, where apps declare collecting data synonymous with user identity as being *not linked* with the users identity (e.g., `Device ID`), or collecting data for the purpose of personalization as *not linked* to the users identity, which would render personalization impossible. Either the documentation given by Apple for the different data types, privacy types, and purposes is incomplete and permits such collection, or a coherence check should be included when a developer uploads a privacy label to warn that such declarations are impossible.

We performed three case studies on Games, Finance, and Social Networking apps, revealing noticeable differences between those app categories. Gaming apps mainly collect tracking and advertisement data; Social Networking and Finance apps do not collect such data over proportional frequent. Moreover, we can confirm the collection of purpose-related data especially for Finance and Social Networking apps. Additionally, we demonstrated that examining single app categories using keyness plots is valuable for multiple reasons: it (1) allows a closer look into specific app categories to assess stereotypes and abnormalities, (2) uncovers similarities or differences, and (3) assists a first exploratory analysis of our large data.

Finally, we analyzed the possible declarations in the context of the GDPR, assessing that any data point collected *linked to the user* or for *tracking* requires consent by the user, unless specific laws permit otherwise. Determining these situations would require a case-by-case and country-by-country analysis.

All data collection is self-declared by the developer and, as far as we know, those declarations are not checked for truthfulness by Apple. Consequently, it is possible for developers to flout their own declarations.

6 Traffic Collection iOS

We took an in-depth look into the Apple privacy labels (Section 3) and what they tell us about self-declared data collection (Section 4 and 5). However, a privacy label is a self declaration, and may not necessarily be correct. We are interested in validating adherence to the labels by collecting and analysing traffic transmitted by an app, to check against the app's privacy labels.

For this, we design a framework to collect traffic on iOS (Section 6.1) and then implement that framework to collect the traffic of 1687 apps (Section 6.2).

6.1 iOS Traffic Collection Framework

Our framework for intercepting network traffic from iOS apps was implemented using a jailbroken iPhone, mitmproxy [15], Frida [11], and SSL Kill Switch 2 [19].

The traffic collection for a single application proceeds in 4 steps: (1) Installing the application, (2) granting permissions to the application, (3) starting the application and intercepting the traffic, and (4) removing the application.

6.1.1 Method

To enable root access to the iPhone, we use the Checkra1n jailbreak [14]. We then install Frida and SSL Kill Switch 2. Finally, we configure the iPhone to use our wireless network and our computer as a proxy, as well as adding the mitmproxy configuration profile in iOS. After the initial preparation for the traffic analysis is done, each app is run sequentially using our methodology.

(1 & 4) Installing and Removing Applications: We install and remove apps automatically using the cfgutil provided by Apple [20]. This allows a computer, connected via USB, to install and remove apps on the connected iPhone.

(2) Granting Permissions: iOS (14.7) uses an SQLite database for permission management[1]. We interact directly with this database to grant the desired permissions. This allows us to set every permission[2] except access to the phone's geolocation. To grant this permission, we inject Frida into the settings app and

1 the location is /private/var/mobile/Library/TCC/TCC.db

2 All known permissions are listed in Appendix 13 Table 5.

grant the location permission directly through the user interface.

(3) Running Applications and Intercepting Traffic: Before running the application, we start our mitmproxy on the experiment computer already configured as a proxy for the iPhone. The corresponding configuration profile has already been installed on the iPhone and consequently, in combination with SSL Kill Switch 2, the certificate of mitmproxy is implicitly trusted, allowing us to intercept and collect encrypted traffic. SSL Kill Switch 2 disables SSL verification as well as certificate pinning.

We do not interact with the network traffic or the app, and are only passive observers, i.e., we perform no-touch traffic collection. After the configured measurement time has passed, we remove the application to stop it.

6.2 Collecting iOS Traffic

In this Section, we specify the applications analyzed and describe the parameters of our experiment.

6.2.1 Collecting iOS Apps

There is no publicly available repository of iOS app IPAs, and an app has to be signed for a user's device.

The 3u application, previously used to get the list of app IDs, provides a download interface storing the corresponding IPA on the local hard drive and registering the app to the user[3]. We manually downloaded the top 100 apps of October 2021 in each category curated by 3u.

After accounting for redundancies across categories, our final dataset consists of 1687 unique applications. We again applied our iOS privacy label process to retrieve the corresponding privacy labels.

6.2.2 Experimental Parameters

We performed our data collection using an iPhone 8 (iOS 14.7.1) with a logged-in Apple user account. Each app was allowed to run for one minute on the phone.

3 A traffic analysis confirmed that 3u is simply an interface for the actual Apple App store.

6.2.3 Honey Data

To make leaked private information recognisable in the network traffic and to simulate an in-use smartphone, we prepared the runtime environment with honey data. Table 4 in the Appendix lists the various honey data points and how they are seeded.

As an application might store information in the clipboard, we need to repeat this step before every app installation to ensure control over the values, via Frida.

7 Observed iOS App Traffic

Before analyzing the collected app traffic, we purged all traffic records of domains we observed while iOS was idling, or that are owned by Apple (apple.com and icloud.com). The remaining traffic records comprised 51232 collected requests with a mean request count of 30.37 and a standard deviation of 51.61 per app.

7.1 Contacted Advertisers and Trackers

We used Easy List and Easy Privacy [9] to check contacted hosts for known advertisers and known trackers, respectively. The most popular detected advertiser and tracker domain is *facebook.com* with 442 apps.

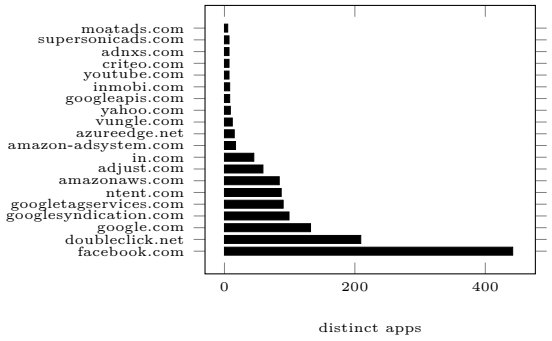
The category contacting advertisement domains the most is lifestyle, and the category contacting trackers the most is lifestyle. Overall, 1085 apps contacted at least one advertiser and 1188 apps contacted at least one tracker.

Fig. 8a and 8b shows the popularity of the top 15 Easy List and Easy Data contacts, respectively.

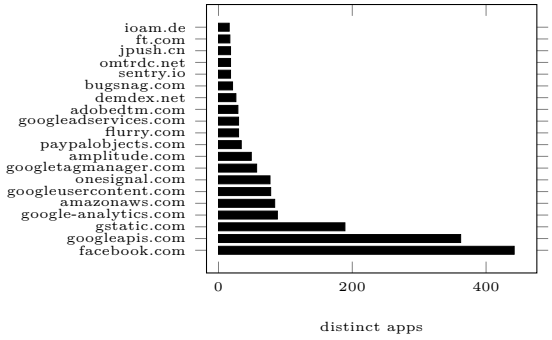
7.2 IDFA Transmission

The IDFA (Identifier for Advertiser) is an iOS-provided value to identify a user across applications and vendors. Even though, starting with iOS 14.5, tracking requires explicit user consent, the API to retrieve the IDFA is still available. However, unless the user explicitly permits tracking, the API returns an IDFA-value consisting completely of zeros [13].

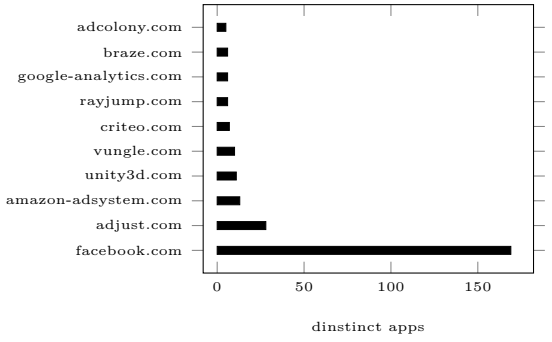
We searched the collected traffic records and found 282 apps transmitting this all-0 IDFA value, which is a clear indicator of these apps' intent to track the user across apps and vendors. On average, 17.77 apps per category transmitted this value with a standard deviation



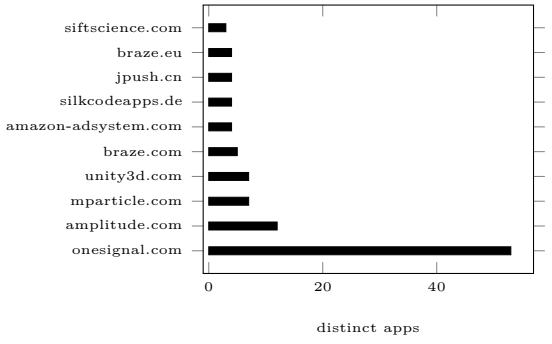
(a) Easy List Contacts (Advertisement)



(b) Easy Privacy Contacts (Tracking)



(c) Number of apps sending at least one request containing the IDFA to the domain.



(d) Domains receiving honey data from apps.

Fig. 8. Domains contacted by distinct apps.

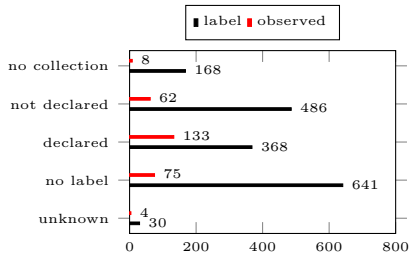


Fig. 9. Apps, aggregated across categories, transmitting the IDFA contrasted with how the apps declared such data collection. The x axis denotes the number of apps.

of 9.58. The category with the most apps transmitting this value was Photo and Video, with 38 apps. Fig. 8c shows the 10 most popular hosts receiving IDFAs, with the most popular being facebook.com. In addition, Fig. 9 contrasts the number of apps declaring collecting the IDFA with the number of apps detected transmitting it.

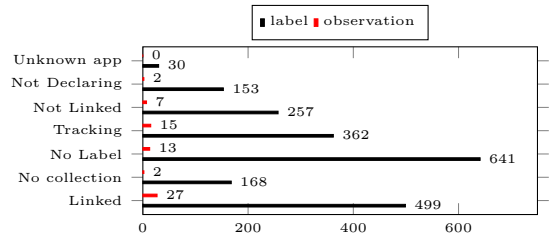
7.3 Honey Data Transmission

We examined the collected traffic for any of our known honey-data strings in plain text. All manually inserted honey data are a random string to ensure that false positives are unlikely. We detected transmission of only a subset of our honey-data points: device name, local IP, location, OS version, and Wi-Fi Name. Fig. 8d shows the 10 most popular domains receiving any honey data (excluding the OS version), with the most popular being onesignal.com.

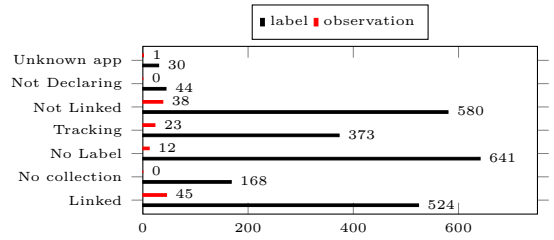
Overall, 276 apps transmitted data not listed in their privacy labels. To compare the honey-data transmission with the apps' self-declared data collection practices, we associated the honey data with privacy label data categories (Table 4). Fig. 10 gives a visual overview of the different honey-data results with 'no collection', meaning that the app declared that it does not collect any data. 'Not declared' means that the corresponding privacy label did not contain the honey-data–associated data category.

7.4 Visible Privacy Consent Form

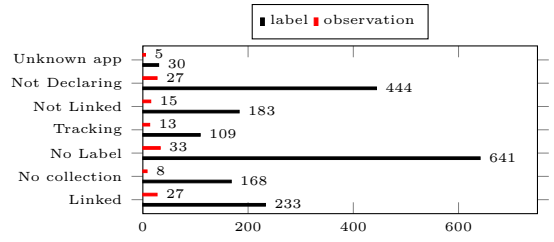
During the traffic measurement, we took a screenshot of each app after 60 seconds. We then checked each screenshot for any of a consent dialogue referencing the GDPR, a privacy policy, or notice. Overall, 192 apps displayed a message fitting our criteria. Furthermore, 7



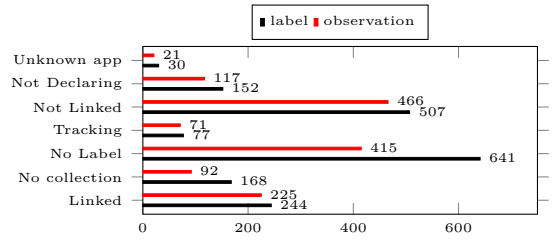
(a) Device Name



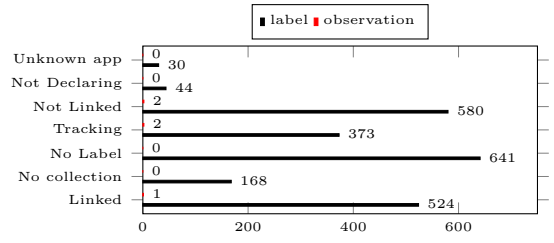
(b) Local IP



(c) Location



(d) OS Version



(e) Wi-Fi Name

Fig. 10. Plots summarizing the detected transmission of honey data. Observations are the apps observed transmitting data. Labels are the apps whose privacy label indicates data collection. The x axis denotes the number of apps.

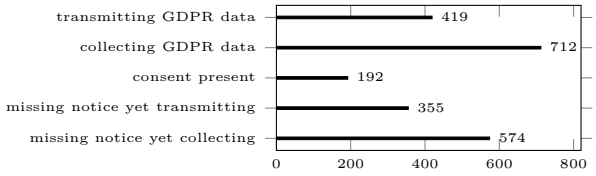


Fig. 11. This plot shows the number of apps that declare collecting data *linked* to the user or for *tracking*, the number of apps transmitting honey data, and the number of apps displaying some form of GDPR or privacy consent message

apps displayed a message notifying us that the iPhone used is jailbroken.

352 apps are transmitting honey data (except the OS version[4]) without some form of consent dialogue or privacy notice in their screenshot.

8 iOS Traffic Discussion

In this Section, we analyse the observed differences between declared data collection and observed data collection, as well as implications of these discrepancies in the context of the GDPR data protection legislation and privacy labels.

8.1 Contacted Advertiser and Tracker

The overall picture for both contacted advertisers and trackers is similar for the leading receivers, with the most contacted tracker and advertiser being Facebook, receiving requests from nearly a quarter of all apps. Facebook is closely followed by Alphabet-owned domains. However, the tail end of the advertiser and tracker domains receiving requests differ markedly with few intersections in their domains.

This shows that both the advertisement and tracking marked are dominated by Facebook and Alphabet, and a variety of smaller companies split whatever is left. The overall distribution of trackers contacted by apps seems to follow a Pareto distribution [24], with a few trackers receiving most of the requests.

Previous work focusing on Android did not show such a high request rate to Google, as these requests were probably indistinguishable from legitimate requests by the operating system [38]. Switching the focus to

4 We exclude the OS version as this is arguably not personal data requiring user consent

iOS thus makes the presence of Google in the mobile tracking market immediately visible. The price, however, is the same problem for any request to Apple, for which we cannot distinguish between legitimate requests and requests used for tracking.

Even though contacting a tracker is not necessarily a GDPR violation, each request leaks the user's IP address plus any additional information contained in the request. Consequently, each such contact risks a GDPR violation [10].

8.2 IDFA Transmission

We observed 282 apps transmitting the IDFA; each such transmission shows that the app intended to track the user. However, 70 apps transmit the identifier without declaring that they collect identifiers for tracking. The lion's share of requests containing the IDFA go to facebook.com, indicating unacknowledged use of the Facebook SDK by the developer.

This is arguably the easiest part of a privacy label to check, as the corresponding value is easy to search for. Additionally, the underlying operating system itself is aware that an identifier is being requested without the user having given permission. Finding such an identifier in the no-touch traffic of an app that does not declare such collection, or the operating system getting a non-permitted API call, should be grounds for further inquiry into an app in any App Store pre-publication check.

The intent for tracking when collecting this identifier is clear, as the developer has access to the Identification for Vendor (IDFV) that is more than sufficient to link data to the user in the app's context. The IDFV is both device- and vendor-specific, and thus allows a vendor to link data to a user without enabling tracking across devices or apps by different vendors.

In the context of the GDPR, regardless of intent, collecting such a unique identifier requires prior consent by the user and only the protection provided by iOS prevents a violation. However, nulling of the value depends on the configuration and version of the operating system. Consequently, a slightly different context would put the app in breach of the GDPR.

8.3 Honey Data Transmission

The most present honey data is the operating system version. Wanting to know the operating system is reasonable, e.g., to understand what devices are still used and thus require the developer's support. Furthermore, the operating system does not directly fit the description of personal data in the context of the GDPR. However, diagnostics is a defined purpose within the Apple privacy labels, which contain a data type for '[a]ny other data collected for the purposes of measuring technical diagnostics related to the app' [3]. Therefore, it is reasonable to expect the developer to declare such a data collection. A relevant portion (153 apps) do not do this.

The remaining detected honey data—device name, local IP, location, and WiFi network name—can be deemed personal data under the GDPR. Given that the most popular domain receiving such requests belongs to Onesignal, a 'self-serve customer engagement solution for Push Notifications, Email, SMS & In-App.' [17] that provides an SDK, and as none of the subsequent popular honey-data–receiving domains could be attributed to a specific app vendor, it is possible that most leakage is due to the undisclosed SDK use. Nonetheless, as we did not consent, we consider each such transmission a violation of the GDPR.

8.4 Visible Privacy Consent Form

In apps' privacy labels, any app self-declaring that it collects either data linked to the user, or data used for tracking, should display a privacy consent dialog. Out of 712 apps requiring such a dialogue, 582 apps do not display a corresponding notice. Fig. 11 visualizes this discrepancy. Enforcing such a compliance would be well within the abilities of Apple as the sole controller of the operating system, by providing a compliant OS-based API to interact with the user to gain consent.

The overall number of apps displaying a dialogue or privacy information is low. Furthermore, our inspection casts doubt on the legality of some of the dialogues, as some only displayed a single 'OK' button or a 'by continuing you agree' notice. This is not considered sufficient, according to common interpretations of the GDPR and is also considered nudging. Nudging has been studied in the context of privacy consent forms and shown to be problematic [46].

8.5 Lessons Learnt

We detected multiple apps that contacted known trackers, transmitted honey data within their traffic, or tried to transmit the IDFA, a unique user identifier. A subset of those apps violated their own privacy labels with that

behaviour. The overall number of disagreeing apps is small but significant, showing that merely declaring privacy labels is not sufficient; enforcement is required as well. We have shown that such enforcement is possible with limited resources by conducting this study.

Furthermore, we analysed the domains receiving tracker IDs and honey data, showing that app vendors rarely collect the data themselves but use third parties. Combined with the lack of privacy consent forms or their proper implementation, these observations paint a picture of developers not caring for what data is actually leaked; this is possibly due to them not actually being aware.

9 Limitations

Our methodology and the opaque ecosystem of data collection and processing entails some limitations in our approach and in the generalisability of our findings.

Traffic Collection We are using a jailbroken iPhone, and apps are able to tell whether or not an iPhone is jailbroken. However, there is no possibility of automating an iPhone without first jailbreaking it. Furthermore, our methodology is strictly no-touch. Depending on the control flow of the application, we do not proceed past any initial message boxes or form fields required to explore and use the app in its full extent. This inevitably leads to the loss of potential network traffic that an app generates while in use. Finally, we limit our traffic collection to one minute after startup and search only for non-obfuscated values. Any value transmitted after the first minute or in an obfuscated fashion is missed. Consequently, we only have a lower bound on the data an app transmits.

Privacy Labels & Honey Data Transmission Apple Privacy Labels declare only what data will be collected, with 'collected' being defined as 'stored longer than required to answer the single request'. This definition leaves the loophole of plausible deniability even if honey data is detected, as the honey data is transmitted but not stored. Consequently, we can make no inference on the usage of a transmitted value; our observation is limited to only its presence in a request. However, we consider the transmission of data to be a strong indicator of further processing.

Visible Privacy Consent Form Visually checking for a privacy consent form or notice comes with inherent drawbacks. Some of our apps were in a language that we do not understand; in those cases, we decided to exclude the app from our study, as our account is localized in Germany and a valid consent form has to be presented in a form that the user can understand [6]. Furthermore, it is possible that the app displays a consent form as soon as the user starts interacting with the app, which would inevitably missed due to our no-touch approach.

10 Related Work

The related work can be split into work primarily concerning iOS privacy [21, 25, 26, 34, 35], and primarily concerning Android privacy [29, 30, 36, 38–40, 45, 48, 49].

For iOS, dynamic analysis has been used to show that almost 80% of apps send and receive data within the first few seconds of launch, and that about half are sharing data with statistical and tracking libraries [35]. Static analysis has been used to detect data flow of private information showing that more than half of all studied apps are leaking the unique ID of the host device [26]. Both of these works shows that privacy information leaks from iOS apps are a threat and that traffic analysis is a powerful approach for detection.

Crossover work, covering both iOS and Android applications, has shown that developers for both iOS and Android are not keen on supplying information on how they collect privacy information, as only a fraction of inquiries were answered, with answers containing misleading information, and a noticeable proportion of vendors were unreachable [34]. Additionally, recent work has also shown that there is no meaningful difference in the data collection behavior of iOS and Android applications [33]. Further work showed that only a small fraction of mobile app users—both Android and iOS—can be considered privacy-aware [25]. Our work shows that, in their current state, Apple's privacy labels are unenforced and may thus contain misleading information, potentially misleading even the privacy-aware users.

Previous work on Android used app automation, traffic collection, and traffic analysis in a fashion similar to ours and revealed personal data transmissions [38, 42], a trend of increased data collection across versions over time [39], and a mismatch between the user's expectations concerning network transmissions and actual transmissions [29]. These findings show that mobile application traffic is rich in informative data, and can be examined for unexpected data leakage.

Orthogonal to our dynamic traffic collection and analysis approach, static analysis and symbolic execution has revealed leaks of sensitive information in An-

droid apps, suggesting potential methodological transfer avenues for future research on iOS [30, 48].

Finally, work on privacy policies showed mismatches between self-declared privacy policy and actual code behaviour [41, 49] and that self-governed privacy policies do not lead to improved privacy preservation in applications [40]. Those works concur with our observation that self-declared privacy labels do diverge from actual observed app behaviour, and our conclusion that enforcement is required.

11 Conclusion

In summary, we comprehensively analyzed the current iOS Privacy Label ecosystem. A systematic survey of the currently deployed privacy labels in the iOS App Store, and an exploratory statistical analysis, allowed us to quantify various stereotypes concerning data-collection patterns, thus providing a basis for further research and more specific investigations.

As our data shows, the majority of apps still transmit personal information. If the introduction of the mandatory labels was hoped to deter privacy-violating actions, such an effect is not noticeable. Especially problematic in this respect are apps in the Games categories.

Next, we put the privacy labels to the test by developing a traffic collection framework for iOS. Using our workflow for automated app installation and execution, we recorded the network traffic of 1687 iOS apps. Before launch, we outfitted the apps with unique honey data, which allowed an unambiguous categorization of the observed network communication into the respective privacy categories.

Our experiment results show that, apparently, no validation of the privacy labels takes place from Apple's side. We were able to identify several apps that ignore their own labels, as they transmit data upon application launch that was not part of the app's privacy declaration. This is both surprising and concerning. It is surprising because our experiment demonstrates that a base-line validation of the labels is clearly feasible. Apple's App Store is known for its comparatively rigorous approval process. Hence, one would reasonably expect that steps similar to our framework take place during app approval.

Our results are concerning for two reasons: For one, it appears that an app's violation of its self-declared privacy label has no negative consequences, and that Apple does not enforce compliance with, or correctness of, the privacy labels. Thus, it is a valid concern that this oversight might be abused more frequently in the future. Our experiments recorded potentially privacy related traffic only during application launch; we did not conduct any interaction with the apps. Thus, we likely only observed the figurative tip of the privacy violation iceberg.

Finally, we checked whether the apps complied with the GDPR, again with sobering results, as numerous apps transmit data that is prohibited under the law without any prior user interaction.

Our work shows that base-line checking of apps against their privacy labels is feasible even with limited resources, and that apparently this is currently not done by Apple, thus leaving developers free to simply lie to their users. Nonetheless, we consider the privacy labels introduced by Apple to be a big step in the right direction, as they provide customers with the ability to choose what data is collected, and for what purposes. However, it is important that the user can actually trust a provided privacy label.

Future Work We touched the topic of non–GDPR-compliant consent forms during our analysis of the app traffic, however, a more in-depth analysis of the consent dialogues and their effects on the data transmission remains to be done. Additionally, our analysis was only no-touch, leading to a significant portion of app traffic not being triggered. Developing a test framework that allows for meaningful app interaction would significantly improve the ability to evaluate the truthfulness of the privacy labels.

12 Availability

Our data and programs are available at https://github.com/Keeping-Privacy-Labels-Honest/Main. If you have any questions you can either use the linked GitHub organization or our email addresses as points of contact.

13 Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 101019206. This project also received funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2092 CASA - 390781972.

References

[1] 22 biggest GDPR fines of 2019, 2020, and 2021 (so far). https://www.tessian.com/blog/biggest-gdpr-fines-2020/. Accessed: 2021-11-24.

[2] 3u - an all-in-one tool for iOS devices. http://www.3u.com/. Accessed: 2021-08-24.

[3] App privacy details on the App Store. https://developer.apple.com/app-store/app-privacy-details/. Accessed: 2021-10-22.

[4] App privacy labels now live on the App Store. https://developer.apple.com/news/?id=3wann9gh. Accessed: 2021-11-01.

[5] Apple privacy statement. https://www.apple.com/privacy/. Accessed: 2021-11-24.

[6] Art. 12 GDPR. https://eur-lex.europa.eu/eli/reg/2016/679/oj. Accessed: 2021-11-30.

[7] Art. 7 GDPR. https://eur-lex.europa.eu/eli/reg/2016/679/oj. Accessed: 2021-08-10.

[8] Data privacy day at Apple: Improving transparency and empowering users. https://www.apple.com/newsroom/2021/01/data-privacy-day-at-apple-improving-transparency-and-empowering-users/. Accessed: 2021-10-22.

[9] Easylist. https://easylist.to/index.html. Accessed: 2022-03-03.

[10] Positionsbestimmung der Konferenz der unabhängigen Datenschutzbehörden des Bundes und der Länder, Punkt 9. https://www.ldi.nrw.de/mainmenu_Datenschutz/submenu_Technik/Inhalt/TechnikundOrganisation/Inhalt/Zur-Anwendbarkeit-des-TMG-fuer-nicht-oeffentliche-Stellen-ab-dem-25_-Mai-2018/Positionsbestimmung-TMG.pdf. Accessed: 2021-11-30.

[11] Frida. https://frida.re/docs/home/. Accessed: 2021-08-10.

[12] The history of the General Data Protection Regulation. https://edps.europa.eu/data-protection/data-protection/legislation/history-general-data-protection-regulation_en. Accessed: 2021-11-24.

[13] iOS API advertisingidentifier. https://developer.apple.com/documentation/adsupport/asidentifiermanager/1614151-advertisingidentifier. Accessed: 2021-11-24.

[14] Jailbreak for iPhone 5s through iPhone X, iOS 12.0 and up. https://checkra.in/. Accessed: 2021-11-24.

[15] mitmproxy. https://mitmproxy.org/. Accessed: 2021-08-10.

[16] Number of smartphone users from 2016 to 2021. https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/. Accessed: 2021-11-24.

[17] Onesignal. https://onesignal.com/. Accessed: 2022-03-04.

[18] Regulation (EU) 2016/679 of the European Parliament and of the Council. https://eur-lex.europa.eu/eli/reg/2016/679/oj. Accessed: 2021-08-10.

[19] SSL kill switch 2. https://github.com/nabla-c0d3/ssl-kill-switch2. Accessed: 2021-08-10.

[20] Use the Apple Configurator 2 command-line tool. https://support.apple.com/guide/apple-configurator-2/use-the-command-line-tool-cad856a8ea58/mac. Accessed: 2021-08-10.

[21] Y. Agarwal and M. Hall, "ProtectMyPrivacy: Detecting and mitigating privacy leaks on iOS devices using crowdsourcing," in *The 11th International Conference on Mobile Systems, Applications, and Services (MobiSys'13)*, 2013.

[22] M. I. Akbas, R. N. Avula, M. A. Bassiouni, and D. Turgut, "Social network generation and friend ranking based on mobile phone data," in *2013 IEEE International Conference on Communications (ICC)*, 2013.

[23] B. Andow, S. Y. Mahmud, J. Whitaker, W. Enck, B. Reaves, K. Singh, and S. Egelman, "Actions speak louder than words: Entity-sensitive privacy policy and data flow analysis with PoliCheck," in *29th USENIX Security Symposium (USENIX Security 20)*, 2020.

[24] B. C. Arnold, *Pareto Distributions*. New York: Chapman and Hall/CRC, 2015. [Online]. Available: https://doi.org/10.1201/b18141

[25] Z. Benenson, F. Gassmann, and L. Reinfelder, "Android and iOS users' differences concerning security and privacy," in *ACM SIGCHI Conference on Human Factors in Computing Systems (CHI'13)*, 2013.

[26] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "PiOS: Detecting privacy leaks in iOS applications," in *The 18th Annual Network & Distributed System Security Symposium (NDSS'11)*, 2011.

[27] P. Emami-Naeini, J. Dheenadhayalan, Y. Agarwal, and L. Faith Cranor, "Which privacy and security attributes most impact consumers' risk perception and willingness to purchase IoT devices?" in *IEEE Symposium on Security and Privacy*, 2021.

[28] P. Emami-Naeini, H. Dixon, Y. Agarwal, and L. F. Cranor, "Exploring how privacy and security factor into IoT device purchase behavior," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, 2019.

[29] D. Ferreira, V. Kostakos, A. R. Beresford, J. Lindqvist, and A. K. Dey, "Securacy: An empirical investigation of Android applications' network usage, privacy and security," in *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec '15)*, 2015.

[30] C. Gibler, J. Crussell, J. Erickson, and H. Chen, "AndroidLeaks: Automatically detecting potential privacy leaks in android applications on a large scale," in *Trust and Trustworthy Computing*, 2012.

[31] M. H. Herzog, G. Francis, and A. Clarke, *The Multiple Testing Problem*. Springer, 2019.

[32] P. G. Kelley, J. Bresee, L. F. Cranor, and R. W. Reeder, "A "nutrition label" for privacy," in *Proceedings of the 5th Symposium on Usable Privacy and Security*, ser. SOUPS '09, 2009.

[33] K. Kollnig, A. Shuba, R. Binns, M. V. Kleek, and N. Shadbolt, "Are iPhones Really Better for Privacy? A Comparative Study of iOS and Android Apps," *Proceedings on Privacy Enhancing Technologies*, 2022.

[34] J. L. Kröger, J. Lindemann, and D. Herrmann, "How do app vendors respond to subject access requests? a longitudinal privacy study on iOS and Android apps," in *The 15th International Conference on Availability, Reliability and Security (ARES 2020)*, 2020.

[35] A. Kurtz, A. Weinlein, C. Settgast, and F. Freiling, "DiOS: Dynamic privacy analysis of iOS applications," in *Technical Report*, 2014.

[36] A. Mylonas, M. Theoharidou, and D. Gritzalis, "Assessing privacy risks in Android: A user-centric approach," in *Risk Assessment and Risk-Driven Testing*, 2014.

[37] K. A. Nguyen, R. N. Akram, K. Markantonakis, Z. Luo, and C. Watkins, "Location tracking using smartphone accelerometer and magnetometer traces," in *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 2019.

[38] T. T. Nguyen, M. Backes, N. Marnau, and B. Stock, "Share first, ask later (or never?) studying violations of gdpr's explicit consent in Android apps," in *30th USENIX Security Symposium (USENIX Security 21)*, 2021.

[39] J. Ren, M. Lindorfer, D. Dubois, A. Rao, D. Choffnes, and N. Vallina-Rodriguez, "Bug fixes, improvements,... and privacy leaks—a longitudinal study of PII leaks across Android app versions," in *Proc. of the Network and Distributed System Security Symposium (NDSS)*, 2018.

[40] I. Reyes, P. Wijesekera, J. Reardon, A. E. B. On, A. Razaghpanah, N. Vallina-Rodriguez, and S. Egelman, ""won't somebody think of the children?" examining COPPA compliance at scale," in *Proceedings on Privacy Enhancing Technologies Symposium (PoPETS 2018)*, 2018.

[41] R. Slavin, X. Wang, M. B. Hosseini, J. Hester, R. Krishnan, J. Bhatia, T. D. Breaux, and J. Niu, "Toward a framework for detecting privacy policy violations in Android application code," in *Proceedings of the 38th International Conference on Software Engineering (ICSE '16)*, 2016.

[42] R. Stevens, C. Gibler, J. Crussell, J. Erickson, and H. Chen, "Investigating user privacy in Android ad libraries," in *Workshop on Mobile Security Technologies (MoST)*, 2012.

[43] D. Sun and W. C. Lau, "Social relationship classification based on interaction data from smartphones," in *2013 IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops)*, 2013.

[44] C. Taylor and A. Marchi, *Corpus Approaches to Discourse: A Critical Review*. Routledge, 2018.

[45] M. Theoharidou, A. Mylonas, and D. Gritzalis, "A risk assessment method for smartphones," in *Information Security and Privacy Research*, 2012.

[46] C. Utz, M. Degeling, S. Fahl, F. Schaub, and T. Holz, "(un)informed consent: Studying GDPR consent notices in the field," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019.

[47] C. Wang, C. Wang, Y. Chen, L. Xie, and S. Lu, "Smartphone privacy leakage of social relationships and demographics from surrounding access points," in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, 2017.

[48] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "AppIntent: Analyzing sensitive data transmission in Android for privacy leakage detection," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security (CCS'13)*, 2013.

[49] S. Zimmeck, P. Story, D. Smullen, A. Ravichander, Z. Wang, J. Reidenberg, N. C. Russell, and N. Sadeh, "MAPS: Scaling privacy compliance analysis to a million apps," in *Proceedings on Privacy Enhancing Technologies Symposium (PoPETS 2019)*, 2019.

Appendix

This Appendix contains additional information and plots concerning the analysis of privacy labels and how to generate keyness plots.

Generating Keyness Plots

Our analysis relied on keyness plots, widely used in linguistics to examine differences between texts by analysing the word counts via χ^2 statistics. As this statistical technique is uncommon in computer science, we give a brief introduction on how to generate and interpret a keyness plot, based on simple fictional data.

Let's assume a fruit basket containing 100 apples, 100 pears, and 100 oranges. Each fruit can either be sweet, sour, or bitter, as shown in Table 3. We want to compare the flavour profile of the apples to the flavour profiles of the remaining fruits.

Table 3. 300 fruits classified by flavour

	Apples	Pears	Orange
Sweet	**20**	**60**	**10**
Sour	**60**	**10**	**40**
Bitter	**20**	**30**	**50**

We now calculate the keyness for the three different flavour attributes. Our chosen metric for keyness is χ^2, calculated from the expected numbers E of apples, pears and oranges with each flavour.

Sweet:

$$\begin{aligned}
\text{distribution} &= \frac{20+60+10}{300} && = 30\% \\
E(\text{apple}) &= 30\% \times 100 && = 30 \\
E(\text{pear}+\text{orange}) &= 30\% \times 200 && = 60 \\
\chi^2_{\text{sweet}} &= \frac{(20-30)^2}{30} + \frac{(70-60)^2}{60} && = 5
\end{aligned}$$

Sour:

$$\begin{aligned}
\text{distribution} &= \frac{60+10+40}{300} && = 36\% \\
E(\text{apple}) &= 36\% \times 100 && = 36 \\
E(\text{pear}+\text{orange}) &= 36\% \times 200 && = 72 \\
\chi^2_{\text{sour}} &= \frac{(60-36)^2}{36} + \frac{(50-72)^2}{72} && = 22.72
\end{aligned}$$

Bitter:

$$\begin{aligned}
\text{distribution} &= \frac{20+30+50}{300} && = 33\% \\
E(\text{apple}) &= 33\% \times 100 && = 33 \\
E(\text{pear} + \text{orange}) &= 33\% \times 200 && = 66 \\
\chi^2_{\text{bitter}} &= \frac{(20-33)^2}{33} + \frac{(80-66)^2}{66} && = 8.1
\end{aligned}$$

The resulting keyness plot (Fig. 12) shows that 'sour' has a high keyness for the set of apples, i.e., apples are more frequently sour than either pears or oranges, whereas 'sweet' and 'bitter' have higher keyness for the other two fruits, i.e., apples are less often sweet or bitter.

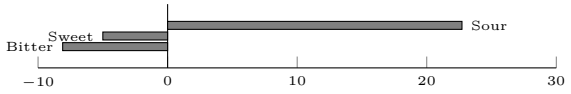


Fig. 12. Keyness plot for apples.

Plots & Tables

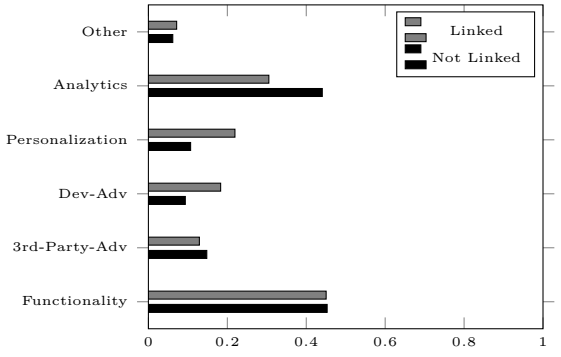
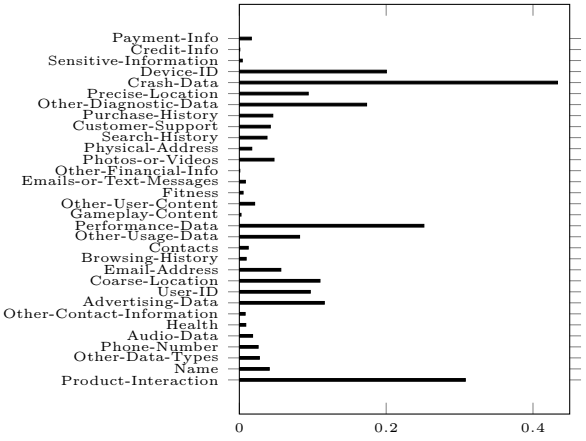
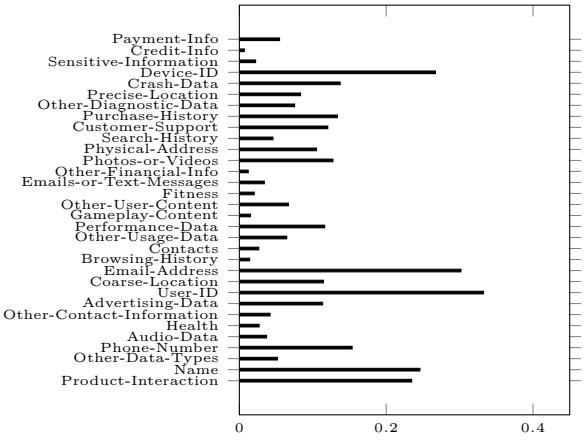


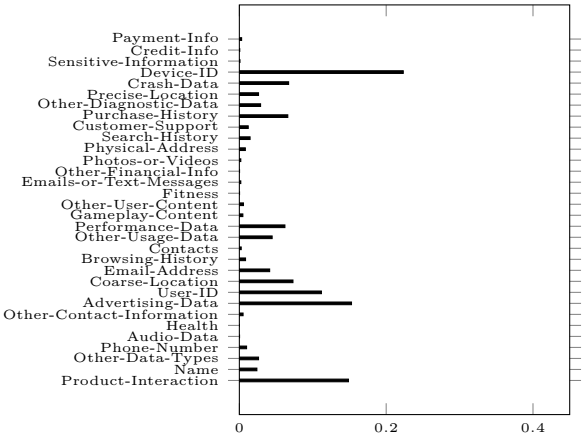
Fig. 13. The proportion of apps collecting *linked* or *not linked* data for a given purpose.



(a) Proportion of apps collecting data types *not linked* to the user.

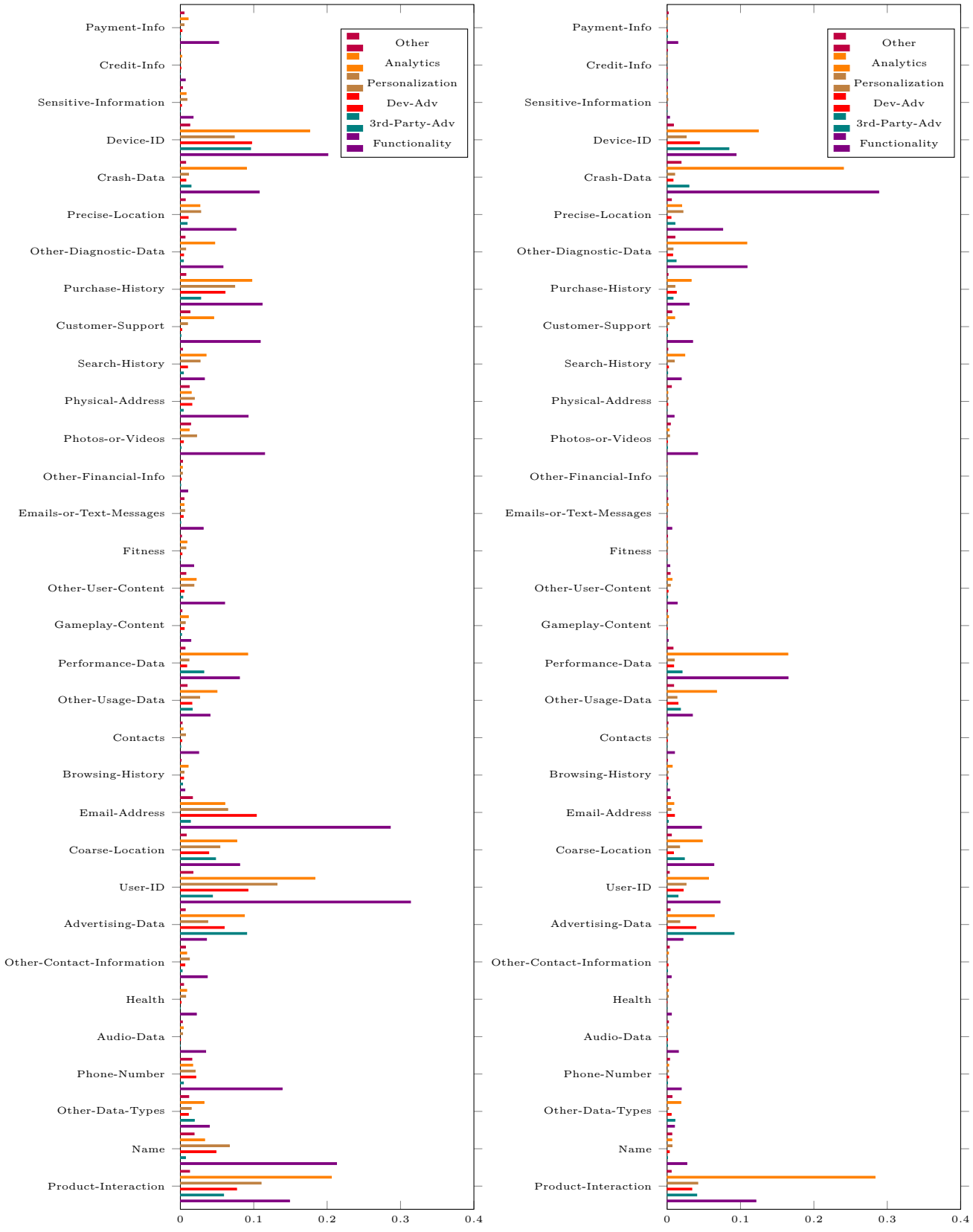


(b) Proportion of apps collecting data types *linked* to the user.



(c) Proportion of apps collecting data types for *tracking*.

Fig. 14. Plots showing the proportion of apps collecting data types split across privacy type.



(a) Proportion of apps collecting data types *linked* to the user split across the purposes.

(b) Proportion of apps collecting data types *linked* to the user split across the purposes.

Fig. 15. Proportion of apps collecting data types split across the purposes.

Table 4. The honey data seeded into the iPhone. Column Value Set describes how the value is set, and Data Category defines the category we attribute the value to. Manually set values are ensured to be unique.

Name	Description	Value Set	Data Category
Contacts	Contacts	manual	Contacts
Location	Device Location	automatically/GPS Sensors	Location
Calendar	Repeating calendar entry	manual	User Content
Clipboard	Data stored in the clipboard	Frida	User Content
Messages	A text message	manual	User Content
Reminder	Regular reminder	manual	User Content
Note	A note entry	manual	User Content
home data	Home data	manual	User Content
wifi	Wi-Fi name	setup	Diagnostics,Location,Identifiers
device name	The name of the device	manual	Identifiers
os version	The OS version	device	Diagnostics
model number	The phone's model number	device	Diagnostics, Identifiers
serial number	The phone's serial number	device	Diagnostics, Identifiers
imei	The phone's IMEI number	device	Diagnostics, Location, Identifiers
wifi-addr	The MAC address of the Wi-Fi	device	Diagnostics, Identifiers
bluetooth addr	The MAC address of the Bluetooth device	device	Diagnostics, Identifiers
local-ip	The phone's IP address	setup	Diagnostics, Location, Identifiers
seid	The SEID	device	Diagnostics, Identifiers

Table 5. This table shows the known permissions configurable via the permission database. The column set denotes wether we granted the permission.

Identifier	Description	Set
Calendar	access to the calendar	✓
AddressBook	access to the address book	✓
Reminders	access to the reminder	✓
Photos	access to the photos	✓
MediaLibrary	access to the media lib	✓
BluetoothAlways	access to bluetooth	✓
Motion	access to motion sensors	✓
Willow	access to home-app	✓
ExposureNotification	permission to collect data required for Covid contact tracing	✓
Camera	access to the camera	✗
Microphone	access to the mic	✗
UserTracking	permission for tracking	✗