IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 17 November 2023, accepted 14 December 2023, date of publication 3 January 2024,
date of current version 11 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3349425

RESEARCH ARTICLE

Sharing is Not Always Caring: Delving Into Personal Data Transfer Compliance in Android Apps

DAVID RODRIGUEZ[1], JOSE M. DEL ALAMO[1], CELIA FERNÁNDEZ-ALLER[2], AND NORMAN SADEH[3], (Member, IEEE)

[1]ETSI Telecomunicación, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[2]ETSI Sistemas Informáticos, Universidad Politécnica de Madrid, 28040 Madrid, Spain
[3]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA

Corresponding author: Jose M. Del Alamo (jm.delalamo@upm.es)

This work was supported in part by the Ministerio de Ciencia e Innovación (MCIN)/Agencia Estatal de Investigación (AEI)/10.13039/501100011033 under Project TED2021-130455A-I00, and in part by European Union "NextGenerationEU"/Plan de Recuperación, Transformación y Resiliencia (PRTR). The work of Jose M. Del Alamo was supported by Spanish "Ministerio de Universidades" through "Movilidad" Sub-Program of "Programa Estatal para Desarrollar, Atraer y Retener Talento," within "Plan Estatal de Investigación Científica, Técnica y de Innovación 2021–2023."

ABSTRACT In an era marked by ubiquitous reliance on mobile applications for nearly every need, the opacity of apps' behavior poses significant threats to their users' privacy. Although major data protection regulations require apps to disclose their data practices transparently, previous studies have pointed out difficulties in doing so. To further delve into this issue, this article describes an automated method to capture data-sharing practices in Android apps and assess their proper disclosure according to the EU General Data Protection Regulation. We applied the method to 9,000 random Android apps, unveiling an uncomfortable reality: over 80% of Android applications that transfer personal data off device potentially fail to meet GDPR transparency requirements. We further investigate the role of third-party libraries, shedding light on the source of this problem and pointing towards measures to address it.

INDEX TERMS Android, compliance assessment, data protection, data transfer, dynamic analysis, GDPR, large language model, personal data, privacy policy, third-party.

I. INTRODUCTION

Data privacy, often identified as data protection, has become a hot topic, gaining increasing attention over recent years. This surge is primarily fueled by growing user concerns, prompting the formulation and update of data protection laws. Among such regulations, the General Data Protection Regulation (GDPR) [1] stands out as the European mandate for data protection with a global impact worldwide [2]. In fact, this regulation has been a model for drafting similar legal provisions in other countries [3].

The GDPR is founded on seven key principles [4]: lawfulness, fairness, and transparency; purpose limitation; data minimisation; accuracy; storage limitation; integrity and confidentiality (security); and accountability. Among them, our study particularly emphasizes the principle of transparency.

The transparency principle mandates that any information and communication relating to personal data processing be presented "*in a concise, transparent, intelligible, and easily accessible form, using clear and plain language*" (GDPR, Art. 5(1)(a)). Furthermore, Article 13 specifies the information that the data controller (i.e., the entity determining the purposes and the means of processing of personal data) must provide to the data subject (i.e., the individual whose data is being processed) when collecting their personal data. In particular, Article 13(1)(e) asserts that the data controller shall specify the "*recipients or categories of recipients of the personal data, if any*". In GDPR terms (Article 4 (9)), a recipient is "*a natural or legal person, public authority, agency or another body, to which the personal data are disclosed, whether a third party or not.*". Third

The associate editor coordinating the review of this manuscript and approving it for publication was Alessandro Pozzebon.

parties are defined in the Article 4 (10) as "*a natural or legal person, public authority, agency or body other than the data subject, controller, processor and persons who, under the direct authority of the controller or processor, are authorised to process personal data*". Therefore, Article 13(1)(e) requires disclosing the identity or categories of any recipient of personal data other than the data controller.

Personal data recipients are particularly abounding in the mobile app ecosystem. Modern software development paradigms call for the integration of online services, exposed through Application Programming Interfaces (APIs), to streamline app development and monetize them. These services offer convenient functionalities and features the app provider does not need to create from scratch e.g., app analytics, identity services, or ads serving, offering significant time-saving and revenue opportunities. To this end, they usually carry out personal data transfers to different recipients. Oftentimes these services are wrapped as code libraries (also known as third-party libraries or Software Development Kits - SDKs), which are packed in the app and delivered jointly with the app's own code into the user device. The popularity of some SDKs has escalated to the point where these libraries now contribute more code to the app than the app developers' own code [5]. Our research specifically focuses on these libraries and their personal data transfers, investigating how their involvement in data processing often remains obscured or inadequately disclosed to the users, thus potentially breaching GDPR transparency requirements.

While SDKs are capable of collecting and transmitting users' personal data to external recipients, it is important to clarify that the responsibility for GDPR compliance is in the app's data controller. When an app is granted permission to access certain resources, all integrated libraries, including these SDKs, automatically inherit these permissions. However, they do not independently determine their use. Consequently, it is crucial for the apps' data controllers to have a comprehensive understanding of the functionalities of these SDKs. They must ensure that their use of these SDKs aligns with GDPR requirements, particularly concerning transparency and lawful processing of personal data. This is essential to mitigate privacy risks for users and to uphold the app's compliance with GDPR mandates.

Official guidelines on transparency under GDPR state that "*the actual (named) recipients of the personal data, or the categories of recipients, must be provided. In accordance with the principle of transparency and fairness, controllers must provide information on the recipients that is most meaningful for data subjects. In practice, this will generally be the named recipients, so that data subjects know exactly who has their personal data. If controllers opt to provide the categories of recipients (either because their identity may regularly change, or because the list would be overwhelmingly long), the information should be as specific as possible by indicating the type of recipient (i.e. by reference to the activities it carries out), the industry, sector and sub-sector and the location of the recipients*" [6]. The same idea is supported by other guidelines provided by the European Commission [7], by the Information Commissioner's Office (ICO) in the UK [8], and by the Court of Justice of the European Union [9]. These official guidelines are summarized in Table 1, where both possible mechanisms —actual name or categories— are presented.

However, disclosing the personal data recipients' categories with the level of detail required by the aforementioned guidelines — including specifics like activity, industry, sector, sub-sector, and location — is notably more challenging and less precise than simply identifying recipients by name. Consequently, the disclosure of such detailed information would be rare. In fact, we manually reviewed 100 privacy policies from applications identified as potentially non-compliant (i.e., transferring personal data to unnamed recipients) during our experiments. Notably, none of them detailed the activity, sector, sub-sector, or location of the unnamed recipients. This omission suggests non-adherence to regulatory guidelines, potentially leading to GDPR non-compliance. In response to these insights, our work is specifically designed to focus on the identification and verification of explicitly named recipients.

Our study aims to shed light on the transparency of personal data transfers in the Android ecosystem and the role of code libraries in this process. To this end, we describe an automated method to capture data-sharing practices in Android apps, assess their proper disclosure according to the GDPR transparency requirements, and understand the source of discrepancies.

To demonstrate the applicability of the method in the wild, we have applied it to 9,000 random Android apps from the Google Play Store, unveiling an uncomfortable reality: over 80% of Android applications that transfer personal data off device potentially fail to meet GDPR transparency requirements. Our findings further suggest that libraries seem to be at the core of the nondisclosure issues.

Aiming to improve adherence to data protection principles in the Android ecosystem, this study describes a fully automated approach that serves multiple stakeholders. Regulators may leverage this method for preliminary, large-scale examinations of privacy practices, subsequently narrowing their focus to apps exhibiting potential non-compliance for detailed investigation. App providers, often unaware of the activities of the SDKs integrated into their applications, can utilize this system to gain insight into SDK behaviors, ensuring accurate disclosures in their privacy policies. Moreover, this research is a component of the autoGDPR initiative (http://autogdpr.org), which aspires to establish a public portal presenting app analyses, thus empowering users with knowledge of privacy implications associated with app usage.

The remaining of the article is organized as follows. Section II presents the related works and contrasts them with our study. Section III introduces and explains the

TABLE 1. Transparency requirements of personal data recipients according to official guidelines.

Required transparency element	Description	Specific requirements
Named Recipients	Naming the recipients refers to disclosing the actual names of the recipients of personal data.	-
Categories of Recipients	If providing named recipients is impossible, controllers may choose to disclose categories of recipients.	• **Type of Recipient**: Description based on the activities carried out by the recipient. • **Industry**: The industry to which the recipient belongs. • **Sector and Sub-sector**: Detailed sector and sub-sector classification of the recipient. • **Location**: Geographical location of the recipient.

method we have developed and its components along with their validation. The method's application on a set of 9,000 Android applications is detailed in Section IV, where the results obtained are also presented. Section V discusses these findings and Section VI concludes the paper.

II. RELATED WORK

Our study identifies personal data recipients in Android apps and assesses whether they are transparently disclosed, further analyzing the extent to which code libraries are implicated in potential compliance issues. This section details those previous works that have touched upon these topics, and how our work compares to them.

Researchers have leveraged static, dynamic, or hybrid techniques to spot personal data transfers in mobile apps [10]. For example, Ferrara and Spoto [11] relied on static code analysis to flag potential personal data leaks in the apps' source code. Jia et al. [12] leveraged dynamic techniques to detect personal data disclosures in network packets. Jia's work could be seen as complementary to ours as we also leverage dynamic analysis techniques to identify personal data transfers, yet we further focus on assessing a transparent disclosure of these data transfers according to GDPR requirements.

Once a personal data transfer is detected the recipient needs to be identified so as to understand if it is disclosed in the app privacy policy. There have been numerous prior works focusing on identifying personal data recipients in both the web and mobile ecosystems [13], [14], [15]. Often referred to as trackers because they specialize in advertising and marketing, these kinds of organizations are the focus of most studies [16], [17]. In a previous work [18], we elaborated a method to reveal the identity of the recipient of a personal data transfer. In addition to the contribution of that work, this paper has the goal of checking whether the apps' disclosures meet the transparency requirements set forth by the GDPR.

The behavior of the code libraries that apps integrate has been the focus of previous research too. Despite most related works concentrating on malware detection [5], [19], [20], [21], some previous studies have attempted to identify code libraries and their personal data leaks. Again, the library identification can be accomplished through static [21], [22], [23], dynamic [24] or hybrid [25], [26] analysis approaches. While these previous works have examined the data transfer behaviors of libraries, demonstrating that some of them pose a significant privacy risk, they have not flagged them as the source of potential compliance issues, as we do in this paper. For the identification of data transfers carried out by libraries, we have leveraged state-of-the-art dynamic analysis techniques that combine the interception of connections in the network with the analysis of stack traces captured during the app execution.

Apps' privacy practices, including the declaration of personal data recipients, have been typically disclosed through privacy policies written in natural language [27]. Machine learning techniques have been widely used to extract information from them. Generally, classifiers are trained with annotated policies [10], which demand a significant time for their coding. Nevertheless, the recent rapid, widespread adoption of chatbots, like ChatGPT based on Large Language Models that do not require specific training, emerges as a promising alternative [28], [29], [30]. Specifically, ChatGPT has demonstrated remarkable performance in processing legal information [31], making it an alternative tool for extracting practices and general information from privacy policies. Our work leverages these state-of-the-art techniques for extracting information from privacy policies, achieving high-performance levels.

Privacy labels, based on the concept of nutrition labels [32], have been recently introduced to disclose privacy practices in mobile apps [33], [34]. Apple introduced privacy labels in the iOS App Store in 2020, compelling app providers to disclose their data practices through a structured schema. A year later, Google introduced the same privacy label concept in its Play Store via its Data Safety Section. Recent studies have shown that privacy labels often contain mistakes [35], [36] and discrepancies with the privacy policies [37], either overstating or understating the apps' privacy practices. In this work, we also analyze the apps' privacy labels to understand if and to what extent they disclose the personal data transfers.

A few previous works have dealt with the assessment of data transfers compliance with GDPR requirements [38], [39], [40]. Razaghpanah et al. [38] analyzed the landscape of tracking services in the mobile ecosystem further discussing GDPR and ePrivacy (the EU legislation on privacy in communications) impact. However, their analysis did not go

deeper into compliance assessment, as they acknowledged "*Our methodology is limited [. . .] and is therefore unable to identify [Advertising and Tracking Services] that are (or, will be) in violation of these regulations*".

Andow et al. [40] and Tan and Song [39] proposed a flow-to-policy check method, which similarly to our work analyzes personal data transfers and then compares the destination entities with those disclosed in the privacy policy. Unlike our proposal, they perform static code analysis to ascertain personal data transfers yet, as stated [39], "*in general, static analysis can obtain more comprehensive data flows, while dynamic analysis can ensure the realness of the detected data flows*". For this reason, and as our method aims to check compliance with GDPR requirements, we have used dynamic analysis to prioritize soundness over completeness.

Furthermore, in our opinion, both papers present limitations to 1) identify named recipients in the privacy policy and, 2) determine the app's data controller. Indeed, both works identify recipients from the destination domain name of the data transfer (e.g., *Adjust* from *http://app.adjust.com*), which precludes detecting organizations that do not resemble their domain name (e.g., *https://teleport.soom.la/* belongs to *ironSource Ltd.*). Andow et al. also follow this strategy for data controller identification by parsing the app package name (e.g., com.mycompany.app). In turn, Tan et al. carry out a manual identification of data controllers, thus obviously limiting the method's scalability. Our method, in contrast, can overcome both, not only accurately identifying the data controller and recipients but also considering whether they are different legal entities, as detailed in the next section.

Therefore, to the best of our knowledge, this is the first work proposing a scalable assessment of the GDPR transparency principle regarding personal data transfers to recipients in Android apps.

III. METHOD

This section details the components of our method to identify personal data recipients in Android apps and assess their transparent disclosure in the apps' privacy policies. Initially, we introduce the method from a high-level perspective, including its context, followed by a thorough explanation of each component's functioning and the validations supporting their performance.

A. OVERVIEW

The method is integrated into a platform that the authors have developed in previous work [41]. This platform automatically downloads applications from Google Play Store including their metadata, privacy policy and privacy labels, installs and runs the apps on real devices with simulated user interactions and events, and intercepts their network communications. To intercept the network connections, this platform utilizes a Man in the Middle (MitM) proxy that enables capturing both HTTP and HTTPS connections. Eventually, the platform searches for already known personal data — extracted from the real devices used — in the body and URL of the intercepted connections, and finally, logs the results (e.g., personal data being transferred, local port used to set up the connection, destination domain, etc.).

Our method receives as input an app's privacy policy and a domain name where the app sends personal information (recipient domain), as observed by the platform. We further modified the platform described above to log the traces leading to a connection setup in the apps, which are also used as inputs to our analysis. The output consists of a disclosure issue flag: positive if the method detects that the recipient has not been adequately disclosed in the privacy policy, or negative otherwise. In the former case, the output also includes the recipient undisclosed and the library initiating the data transfer.

Figure 1 shows the method's main modules, which are responsible for 1) Identifying the personal data recipient (i.e., Recipient Analyzer), 2) assessing their proper disclosure in the app's privacy policy (i.e., Disclosure Checker), and 3) identifying the library triggering the data transfer (i.e., Library Analyzer).

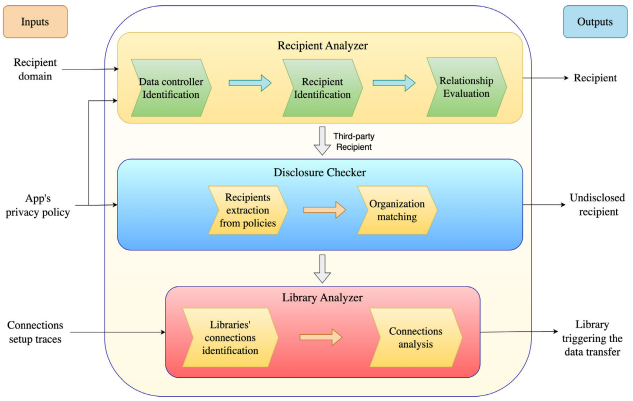


FIGURE 1. Method overview.

The Recipient Analyzer allows determining if the app's data controller and the personal data recipient are actually the same legal entity. The Disclosure Checker evaluates whether the app's privacy policy transparently declares the transfer of personal data to the recipient. The Library Analyzer identifies the library triggering the transfer of personal data to the recipient. Thus, it helps us understand if potential non-compliance issues are related to the use of these libraries.

B. RECIPIENT ANALYZER

This pivotal component aids in determining whether a specific personal data transfer targets the data controller or a recipient. It receives the app's privacy policy and the transfer destination domain, and performs three separate steps: 1) Apps' data controller identification, 2) Recipient identification, and 3) Data controller-to-recipient relationship determination.

1) DATA CONTROLLER IDENTIFICATION

The identification of the app's data controller can be achieved by processing the app's privacy policy text. This is the

most reliable way to identify the organization responsible for the specific app under analysis since the GDPR regulation requires to disclose the data controller. For this task, we utilized the ChatGPT API, specifically the *gpt-3.5-turbo* model, using the *gpt-3.5-turbo-16k* model for lengthy policies.

This step automates a prompt inquiring about the data controller, where the privacy policy is also included in the prompt. According to our tests, the optimal prompt requests the data controller's name for the policy, or NONE if it is unknown (not disclosed). The used prompt is shown in Listing 1.

```
Which company do you think this privacy policy
belongs to? Please ONLY name it. Answer NONE if
you do not know.
PRIVACY POLICY BEGINS HERE:

# Privacy policy content #
```

LISTING 1. Prompt inquiring about data controller in a privacy policy.

This prompt is optimized to facilitate the model comprehension of key information. Capitalized words are used to indicate where the attention should be focused on, as it has been empirically observed to improve the results. The prompt also specifies where the privacy policy begins, thereby avoiding information confusion.

We validated this step on a set of 50 random applications. We manually checked their policies discarding eight non-English texts. Table 2 details the Data Controller Identification performance metrics.

TABLE 2. Performance metrics for the data controller identification and recipient identification steps.

	Data Controller Identification (chatGPT)	Recipient Identification (ROI)
Precision	97.14%	95.71%
Accuracy	88.09%	67.00%
Recall	89.47%	69.07%
F1-score	93.15%	80.24%

2) RECIPIENT IDENTIFICATION

When intercepting personal data transfers, our platform obtains the transfer contents and the destination domain. However, to determine if the transfer is made to a recipient as for GDPR terms, we first need to know the organization owning the destination domain. We leveraged our previous work on a "*Receiver Organization Identifier*" (ROI) tool [18] to this end.

Briefly, this tool uses Selenium and a web search engine to find the privacy policy governing a web domain. It then checks if the text found is indeed a privacy policy, and extracts the data controller's identity using the SpaCy library [42] for the entity recognition process. This method partially meets the goal of the *Data Controller Identification* described above, although it also includes a thorough search process to find the privacy policy governing the target domain. Besides, it applies SpaCy for the data controller identification instead of ChatGPT, as the latter was unavailable at the time of ROI development. The precision of ROI in identifying the organization is 95.71% (Table 2), close to the metric obtained using ChatGPT, although clearly favoring the latter.

3) RELATIONSHIP EVALUATION

Determining the role of an organization as a recipient of personal data transfers involves discerning if the organization receiving the personal data is other than the data controller. Previous works [40], [41] compared the target domain name with the mobile application's name to that end. We significantly improve this process by using their comparison only as a first step. If no (apparent) similarities are found between both fields, ChatGPT is used once again to determine if both the data controller and recipient point at the same entity, with the prompt in Listing 2.

```
Are #company1 and #company2 the same company?
Please answer only with YES or NO. They would be
the same company if they refer to the same legal
entity. They cannot be considered the same company
 only because they belong to the same corporate
network.
```

LISTING 2. Prompt inquiring if both companies are actually the same entity.

ChatGPT serves the specific purpose of dealing with different entities where string matching could lead to false negatives (e.g., renamed companies or corporate designations). This component outputs a boolean answer, allowing us to categorize the recipient as either the data controller or a recipient. We utilize the Disclosure Checker component to verify if the recipients are adequately disclosed in the privacy policies according to the GDPR guidelines.

4) PIPELINE VALIDATION

The Recipient Analyzer pipeline was validated with a dataset of apps and recipients observed in personal data transfers from previous experiments we carried out. Our dataset included 50 destination domains coded as data controllers and 50 destination domains coded as recipients. The codification of a domain as data controller or recipient for an app required manual checks via Crunchbase [43]. The method's performance is detailed in Table 3.

C. DISCLOSURE CHECKER

This component assesses whether the recipients of personal data transfers are transparently disclosed in an app's privacy

TABLE 3. Performance metrics for the recipient analyzer and the disclosure checker components.

Performance metric	Recipient Analyzer	Disclosure Checker
Accuracy	88.00%	95.00%
Precision	84.91%	100.00%
Recall	91.84%	86.36%
F1-score	88.24%	92.68%

```
The following text is a privacy policy of an
Android application.

# Privacy policy content #

End of the privacy policy.

Which third-party companies, service providers or
partners are described in the privacy policy?
```

LISTING 3. Prompt inquiring about the recipients described in the privacy policy.

policy. To this end, we leveraged ChatGPT again to extract from apps' privacy policies the recipients that may receive personal data transfers. The ChatGPT prompt used for this is shown in Listing 3.

We process the output set to remove unnecessary business designations (e.g., Corp or Inc). Afterward, we check whether the organizations observed to receive personal data are on this set to determine if the transfers are explicitly disclosed as per GDPR requirements.

Like the previous pipeline, this one underwent a validation using an untapped dataset consisting of 60 randomly chosen privacy policies where the recipients disclosed were coded by one of the authors. The performance in identifying the recipients disclosed in the privacy policy is detailed in Table 3. The remarkable precision highlights the method's capability to correctly identify when a privacy policy transparently discloses the identity of a personal data recipient.

D. LIBRARY ANALYZER

This component aims to discern whether a library initiated a personal data transfer to a recipient. This will allow us to match transparency disclosure issues in privacy policies to libraries, thus shedding light on potential sources of compliance issues.

This method leverages Frida [44], a dynamic analysis tool used to instrument mobile applications and injects new code into a running process. The app behavior can thus be traced and modified at runtime. We capitalized on Frida to capture network connections established by an app where a socket is set up through the Android (standard) platform API. Thus, when a connection is set up its metadata is logged, including the destination domain, the local port used in the device, and the call stack trace. In turn, our MitMProxy also intercepts the device connections in the network and logs the associated metadata (destination domain, local port), reads their content (even if encrypted), and searches for personal data transfers. Thus, we can easily associate the connections identified with the MitMProxy with those logged by Frida by matching their local (origin) ports. Figure 2 depicts this process.

The benefit of matching connections captured with both techniques is that we enrich our knowledge on the data transfer with information on the source code that established these connections, as for the stack traces obtained with Frida. As a result, we are able to understand if a library is the source of issues in the data transfer disclosures.

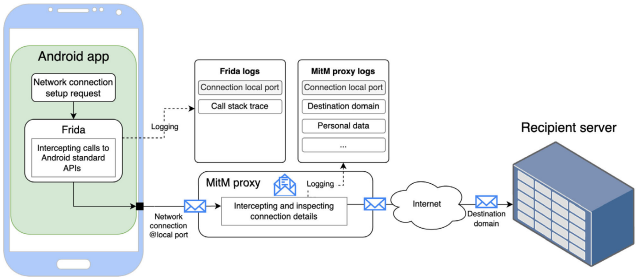


FIGURE 2. Data transfers interception with Frida and a MitM proxy.

1) UNDERSTANDING STACK TRACES

To identify libraries in traces, we first need to understand how a trace is structured. Figure 3 shows an example of a trace, where a connection being setup through the Socket.connect method of the Android standard API is at the top of the trace, as this was one of the methods we monitored. Conversely, the method that triggered the connection setup appears at the bottom of the stack trace, in this case, the *Thread.run* method of the Android standard API. In between, two main pieces of code are found, as for the package names of the classes involved: *com.android.okhttp* belonging to the OKHttp library [45], widely used in Android for handling HTTP and HTTP2 client-side communications; and, *com.my.tracker* belonging to the myTracker library [46], an analytics library.

Upon analyzing the various cases we have encountered in the traces we logged, we observed that oftentimes library methods are the only ones involved in the connection setup leading to a data transfer. That is, the Android standard API methods – i.e. android/dalvik/java/javax namespaces, the JetPack library ones – androidx namespace, or any non-library code was not invoked. In these cases, we consider that it wasn't the app who deliberately initiated the connection since it involved only the library code (despite being legally responsible at all moment for their app's behavior).

2) LIBRARIES IDENTIFICATION

The search for libraries in the stack traces requires knowing their package names, e.g., com.my.tracker in Figure 3. For this purpose, we departed from the Google Play SDK Index [47], which lists the 129 most popular commercial libraries on Google Play.

We leveraged the Maven Central Repository [48] to expand this initial set as it contains a large number of commonly used libraries in the Android ecosystem. To this end, we started from 55,444 stack traces we logged with Frida and, after eliminating duplicates we were left with 18,011 unique package names. Among them, we identified 672 first-party package names (i.e. presumably developed by the data controller) and 2,385 third-party ones (developed by other entities) already found in the Google Play SDK Index. Therefore, our unassigned sample was reduced to 14,954 package names. Among these, numerous package

```
java.lang.Exception
    at java.net.Socket.connect(Native Method)
    at com.android.okhttp.internal.Platform.connectSocket(Platform.java:182)
    at com.android.okhttp.internal.io.RealConnection.connectSocket(RealConnection.java:145)
    at com.android.okhttp.internal.io.RealConnection.connect(RealConnection.java:116)
    at com.android.okhttp.internal.http.StreamAllocation.findConnection(StreamAllocation.java:186)
    at com.android.okhttp.internal.http.StreamAllocation.findHealthyConnection(StreamAllocation.java:128)
    at com.android.okhttp.internal.http.StreamAllocation.newStream(StreamAllocation.java:97)
    at com.android.okhttp.internal.http.HttpEngine.connect(HttpEngine.java:302)
    at com.android.okhttp.internal.http.HttpEngine.sendRequest(HttpEngine.java:245)
    at com.android.okhttp.internal.huc.HttpURLConnectionImpl.execute(HttpURLConnectionImpl.java:465)
    at com.android.okhttp.internal.huc.HttpURLConnectionImpl.getResponse(HttpURLConnectionImpl.java:411)
    at com.android.okhttp.internal.huc.HttpURLConnectionImpl.getResponseCode(HttpURLConnectionImpl.java:542)
    at com.android.okhttp.internal.huc.DelegatingHttpsURLConnection.getResponseCode(DelegatingHttpsURLConnection.java:106)
    at com.android.okhttp.internal.huc.HttpsURLConnectionImpl.getResponseCode(HttpsURLConnectionImpl.java:30)
    at com.my.tracker.obfuscated.r.a(Unknown Source:50)
    at com.my.tracker.obfuscated.x.a(Unknown Source:15)
    at com.my.tracker.obfuscated.x.a(Unknown Source:0)
    at android.activity.c.run(:7)
    at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor. java:1167)
    at java.util.concurrent.ThreadPoolExecutor$worker.run(ThreadPoolExecutor.java:641)
    at java.lang.Thread.run(Thread.java:923)
```

FIGURE 3. Example of stack trace invoking the Socket.connect method.

names had been deliberately obfuscated, so their filtering left us with 6,626 package names. They were searched in the Maven Central Repository, resulting in 152 new libraries being identified. This takes us from the 129 libraries initially reported in the Google Play SDK Index to 281 (eliminating duplicates), translating to a dataset increase of 117.83%.

We applied the library identification step on a set of 20,125 traces coming from old experiments. Interestingly, we observed that 58% of the personal data transfers to recipients are initiated by libraries identified in our dataset, giving an idea of the prevalent role of these libraries in personal data transfers to recipients.

IV. EVALUATION

In this section, we apply the proposed method to the analysis of the personal data transfers to recipients and their disclosure through privacy policies on a randomly selected set of Android applications. In a bid for a more profound understanding, we compare these data transfers with those made by libraries. Finally, we further inspect the apps' privacy labels to discern if this novel disclosure mechanism of disseminating privacy practices proves more reliable.

A. EXPERIMENT DESIGN

We started by randomly selecting 9,000 apps from the Google Play Store. Figure 4 shows the distribution of these apps per category, number of downloads, and average users' ratings. We leveraged our platform to download the apps and their privacy policies and labels, install and run the apps, capture the connections they made during execution, and analyze their contents for detecting personal data transfers. We used five Redmi10 mobile devices running Android 30 for the apps' execution.

The execution of the applications resulted in 202,088 successfully intercepted connections, where we observed 23,840 connections transferring personal data off the device in 4,335 applications (48.17%). The quantity and type of personal data transferred can be observed in Table 4.

Surprisingly, unsecured HTTP connections (3.25%) are still being established by a few apps, some even carrying personal data such as the Google Ad ID or even the precise location, posing a severe privacy risk for their users.

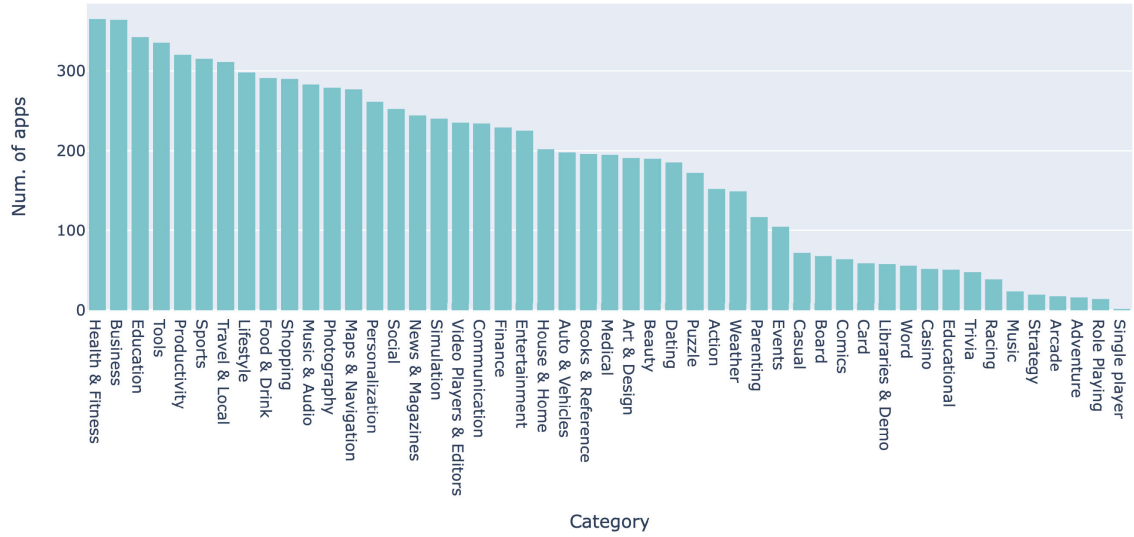
TABLE 4. The type and count of personal data flows captured.

Data type category	Data type description	No. connections (%)
Device_Model	Device model name	22,004 (92.3%)
Google_Ad_ID	Google unique device identifier for advertising purposes	10,846 (45.5%)
Build_No	Build number of Android software version	5,822 (24.4%)
Fingerprint	Unique device identifier based on multiple hardware and software details	1,540 (6.5%)
Router_Wifi_BSSID_Close	Name of near Wifi routers	164 (0.7%)
Router_Wifi_MAC	MAC address of the router connected to the device	136 (0.6%)
Device_location_coarse	Approximate device location	112 (0.5%)
Device_location	Precise device location	106 (0.4%)
Router_Wifi_BSSID	Name of the router connected to the device	82 (0.3%)
Kernel_Version	Android Kernel version	71 (0.3%)

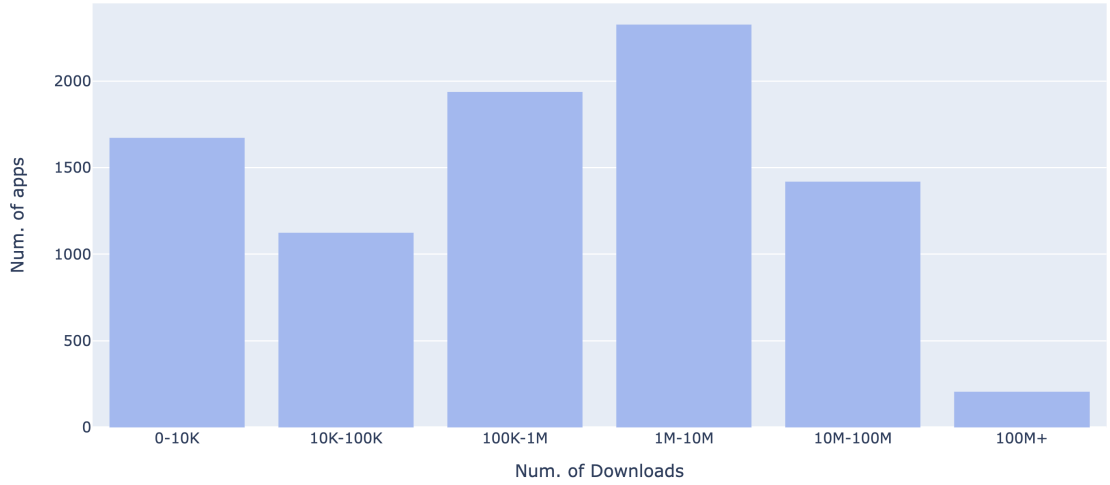
In the following section, we will analyze the recipients of these personal data transfers and the level of transparency we observe in the apps' privacy disclosures.

B. APPS TRANSPARENCY ASSESSMENT

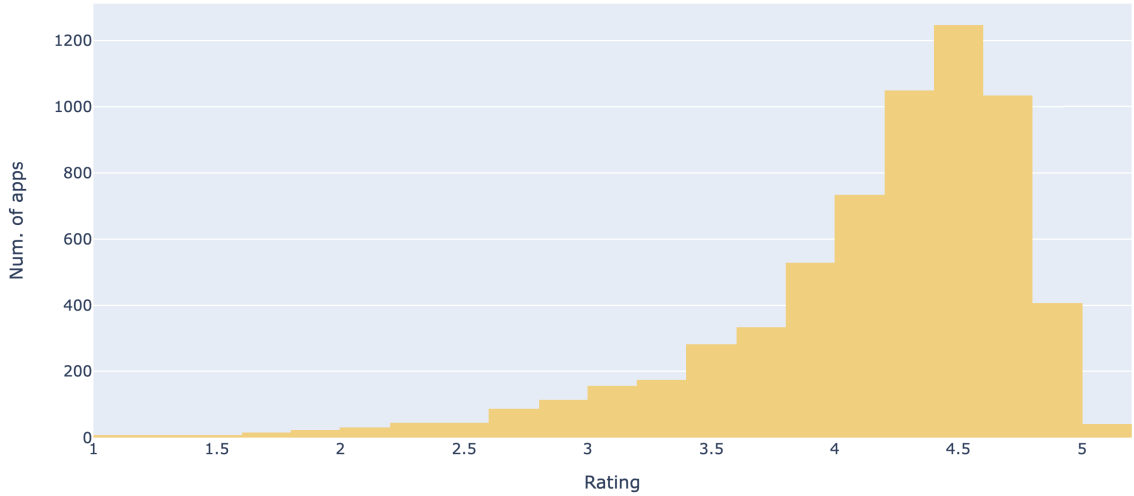
Building on the method detailed in section III, we have been able to identify the recipients of personal data and verify if the apps' privacy policies disclose them adequately.



(a)



(b)



(c)

FIGURE 4. Dataset distribution based on a) categories, b) number of downloads, and c) rating.

We successfully pinpointed the recipient's identity in 17,340 of the connections carrying personal data (corresponding to 3,621 apps). This resulted in 206 distinct recipients, with Google (24.86% connections), Meta (17.93%), and Unity (7.66%) being the most prominent recipients of personal data transfers. Notably, the top-10 identified recipients received the 71.89% of personal data transfers, which is a clear indication of the data collection concentration among very few participants in the Android ecosystem.

We employed the *Data Controller Identification* component to identify each app's data controller by looking into the app's privacy policy. Throughout the process, we identified the data controllers of 1,536 apps responsible for 8,232 data transfers. A significant 25.94% of the supposed app's privacy policies were not actually privacy policies (e.g., they were landing pages instead) or were in languages other than English, leading us to discard them. Interestingly, we also observed that a considerable 22% of apps in our dataset are not transparently declaring or identifying the data controller in their privacy policies, which is a requirement according to GDPR since they are processing personal data (as observed in the connections we intercepted). This points to potential compliance issues by these apps.

Upon identifying the apps' data controllers and the recipients of the data transfers, the *Relationship Evaluation* component determined if these data transfers ended up in the data controller or a different recipient. We were able to determine this relationship in 8,220 (1,536 apps) of the 8,232 connections, noting that 95.4% of the personal data transfers were made to other recipients (1,510 apps). The top recipients were again Google (23.90%), Meta (16.85%), and Unity (9.55%).

We then employed the *Disclosure Checker* component to assess if the apps' privacy policy transparently discloses the data transfers to the identified recipients. This component revealed that 1,225 (81.12%) of the 1,510 apps where the privacy policy could be evaluated failed to disclose the personal data recipients according to the GDPR transparency requirements. Figure 5 shows the 20 recipients that most often are not disclosed by the apps' privacy policies, meaning that these apps' users (in our dataset there are apps with more than one billion downloads) are not informed about who their personal data are transferred to.

These results depict a worrying situation as they show that personal data transfers, while abundant and concentrated in a few organizations, are seldom disclosed to data subjects in the apps' privacy policies. Next, we delve into this issue to understand if the use of libraries may be related to these problems.

C. IDENTIFYING LIBRARIES' TRANSFERS

We further analyzed the source of the connections transferring personal data where the data controller could be identified. This led us to identify the code triggering 6,596 data transfers to other recipients.

Table 5 shows the top 10 recipients whose libraries transfer personal data off the device. It's clear that Google's libraries are the most frequently used, with Google Mobile Services (*com.google.android.gms*) occupying the first place and Firebase Services (*com.google.firebase*) taking the fourth.

TABLE 5. Top-10 recipients whose libraries transfer personal data off the device.

Package name	Library provider (Recipient)	% of connections
com.google.android.gms	Google LLC	25.92%
com.facebook	Meta Platforms, Inc.	9.67%
com.unity3d	Unity Technologies	8.00%
com.google.firebase	Google LLC	5.62%
com.flurry	Flurry	4.52%
com.safedk	AppLovin	3.40%
com.ironsource	IronSource	2.06%
com.adjust.sdk	Adjust	1.76%
com.mbridge.msdk	Mintegral	1.64%
com.inmobi	InMobi	1.56%

We then cross-checked this data with the undisclosed data transfers, to understand where the source of the issues is. We found that the libraries are establishing a worrying 73.68% of the undisclosed data transfers. Again, Google's libraries take first place with 23.57% of undisclosed transfers, followed by Unity's (14.52%) and Meta's (13.57%) in second and third places, respectively. These libraries send personal data like device model, Google advertising ID, and software build number to the different recipients.

D. DATA SHARING DISCLOSURES IN PRIVACY LABELS

We have further analyzed the apps' privacy labels to understand if this new form of privacy disclosure reflects the data-sharing practices of apps. It should be noted that privacy labels present severe limitations in meeting GDPR transparency requirements. As advanced by Novovic [49] the privacy labels "*cannot convey the mandatory obligations required by the GDPR*", being only a complement to other disclosure means such as privacy policies. Indeed, privacy labels allow disclosing the type of data shared with a recipient and the purpose, but their design does not allow disclosing the recipient identity. However, their analysis is still useful to let us know about the apps' awareness of the undergoing data sharing.

To this end, we departed from the 1,510 applications where personal data transfers to recipients were observed and found privacy labels for 1,266 (83.38%) of them. Although privacy labels are mandatory for apps updated since July 20, 2022, we observed that 11 of the 238 (4.62%) apps that do not include privacy labels have been updated after the deadline, which is a potential compliance issue with the Google Play Store policy.

After comparing the data transfers to recipients with the data sharing disclosures in the apps' privacy labels,

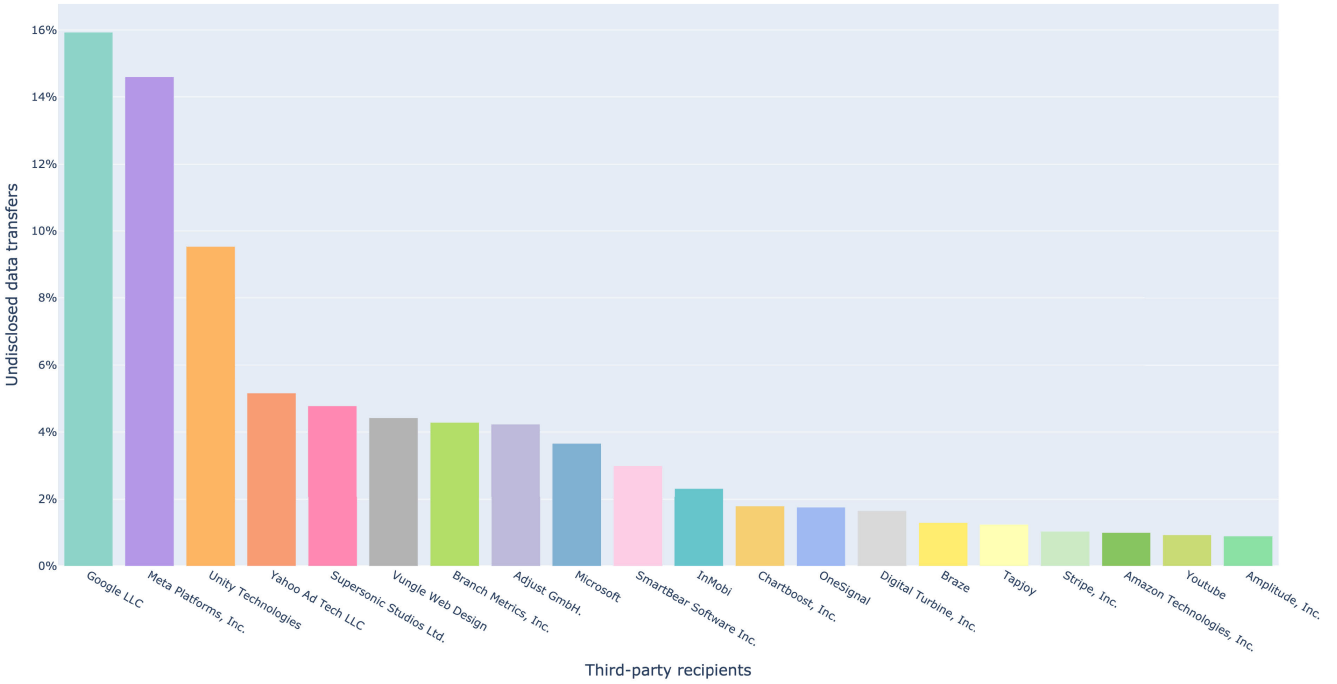


FIGURE 5. Top-20 recipients not being transparently disclosed in the apps' privacy policies. The Y-axis shows the rate of undisclosed data transfers for each recipient over the total amount of undisclosed data transfers.

we observed that 420 (33.17%) apps sent personal data to recipients without disclosure in the labels, which is a sign of these apps' unawareness of their data sharing practices. Notably, all of them were observed sending the *Device or other IDs* without declaring them, except one of them that was found sending the user's *Precise location*.

Interestingly, after analyzing the libraries responsible for the connections undisclosed in the privacy labels, significant differences were observed. The top-five libraries, in descending order, consist of *com.unity3d*, *com.facebook*, *com.safedk*, *com.flurry*, and *com.google*.

A comparative analysis of these 420 applications that fail to disclose the data transfer through their labels paints a dire picture, as 317 of them (75.48%) present severe issues: They neither adhere to their labels disclosures nor do they correctly declare their data-sharing practices in their privacy policies. This presents a dual risk for users, who are left with no apparent way of discerning that their personal data is being dispatched to other organizations. This revelation underscores the magnitude and urgency of the transparency challenges that must be addressed.

On the other hand, these results also indicate that more than half (668 apps, 54.48%) of the apps do acknowledge sharing data in their privacy labels, though they do not disclose the recipient identity as the labels do not allow for it. This finding suggests that most apps are aware of the information being shared, yet they still struggle to disclose it in their privacy policies properly. One possible explanation yields in the features provided by development tools and the Play Store itself, which leverage the information available in the libraries' manifest files to warn developers about the permissions requested by the libraries they integrate, thus calling their attention to disclose them. Unfortunately, these tools do not yet support the automated extraction of finer details such as the entities with which the libraries would be sharing information, which might be used to warn data controllers to disclose the data-sharing practices of their applications transparently.

V. DISCUSSION

The Popularity of an App is Not an Indicator of Greater Transparency: We analyzed if popular applications, according to their number of downloads, better disclose their data-sharing practices. To that end, we computed the rate of disclosing apps over the number of apps in each group in our dataset (Figure 4b). The results did not reveal any particular group of applications as more transparent than the others based on this feature. Nevertheless, it's undeniable that the reach of these apps varies depending on their download counts, posing the most popular apps a greater risk due to their broader user base.

For example, the application with the highest number of downloads in our dataset, com.lenovo.anyshare.gps, fails to disclose the recipients of the personal data it shares. This application, with over a billion downloads on the Google Play Store, has been observed to send the Google advertising ID to Meta Platforms, Inc., Adjust GmbH, and AppsFlyer. However, the app's privacy policy remains ambiguous, indicating that they may share data "*With advertisers and marketing partners in order to display advertisements on our App and support our business, to show how many users of the App have clicked or viewed an advertisement,*

and third-party measurement companies for the purposes of measurement, analytics, engagement technologies and optimization of our Services''. According to the transparency guidelines [50] provided by the European Commission, these privacy policy statements fail due to leaving room for different interpretations and using ambivalent terms (e.g., engagement technologies and optimization of our Services).

Another intriguing case is the *flipboard.app*, which has accumulated over 500 million downloads on the Google Play Store. This application has been spotted transmitting personal data to Adjust GmbH., Microsoft, InMobi, and Upcraft. The Google advertising ID is sent to Adjust and InMobi via the libraries of *com.adjust.sdk* and *com.inmobi*, respectively. The app's privacy policy indicates that personal data can be sent to third parties, including advertising partners. However, none of these organizations are explicitly mentioned, nor activity, sector, sub-sector or location of these recipients are disclosed along with the categories.

Apps' Providers Lack Proper Support to Disclose Their Data-Sharing Practices Accurately: Libraries are at the core of 82% of undisclosed data transfers. Thus, we have checked the websites and the Google Index SDK page for the libraries shown in Table 5 to understand if they properly disclose their data practices. Websites for mainstream libraries generally provide extensive documentation on their privacy practices, including the information they collect and share and, even in some cases, information on how to complete the app's privacy labels (e.g., details on the Unity3D library for Android can be found at https://docs.unity.com/ads/en-us/manual/ImplementingDataPrivacy). On the other hand, the information available at the Google Index SDK is scarce, as it only provides information on the permissions requested by each library (basically, the information available in the library's manifest file), lacking a link to a privacy policy, details on the personal data recipients' identity, or the data collected/shared by the library. As a result, while the information on the libraries' privacy practices is available, it remains scattered, which reduces the ability of data controllers to integrate and disclose it properly, resulting in more than 80% of apps failing to disclose their personal data recipients.

In contrast, our findings suggest that most apps' data controllers are aware of the information being shared, as two-thirds of them managed to report that on their apps' privacy labels. One possible explanation for this different behavior yields in the features provided by development tools and the Google Play Store itself, which leverage the information available in the libraries' manifest files to warn developers about the permissions requested by the libraries they integrate, thus calling their attention to disclose them. Unfortunately, these tools do not yet support the automated extraction of finer details such as the organizations with which the libraries would be sharing information, which might be used to warn developers to disclose the data-sharing practices of their applications properly.

Large Language Models are a Useful Tool for Privacy Policy Analysis: To date, extracting practices from privacy policies has largely leaned on Artificial Intelligence and Natural Language Processing, often necessitating annotated datasets to train Machine Learning models. Crafting these annotations demands both legal and technical expertise. Large Language Models, being trained on vast and varied data, offer coherent responses to prompts, eliminating the need for specific annotation or retraining for each policy feature extraction. Remarkably, ChatGPT comprehends intricate queries, synthesizing information from its extensive training data. For example, in this study, we have noted how it can tailor its responses based on interrelated companies within a group. This represents a substantial time and effort saver, bypassing the technical hurdles of creating Named Entity Recognition components, cross-referencing information from business databases, and the requisites of Natural Language Processing to generate a comparable output.

However, ChatGPT's training posed a hefty computational and temporal demand, constraining its data to 2021 and resulting in significant outdatedness. This proves problematic when discerning ever-evolving business affiliations, potentially leading to inaccurate outputs. This Large Language Model has advanced its ability to understand and retain information from lengthy prompts, now accommodating up to 16k tokens. Larger prompts also come with higher economic (via paid API) and computational (in terms of time) costs, which could affect the quality of results. Newer GPT models exhibit improvements, but their increased model size still restricts their usage and prompt length (128k tokens), with significantly higher prices. Yet, our experimental tests reveal a heightened attention span and a clearer distinction between the query's essence and the actual privacy policy passed as input. Notably, ChatGPT operates non-deterministically by default, meaning the same input might yield varied outputs over time. To address this problem, we are testing temperature and seed parameters to achieve deterministic outputs consistently. Our preliminary tests show that GPT-4 offers more consistent results over time, substantially minimizing these constraints. Using GPT-3.5 solidifies Large Language Models as a robust alternative to ''traditional'' privacy policy extraction techniques. Intriguingly, this is just the tip of the iceberg; with each iteration, its performance only seems to soar, promising even more refined outputs in future versions.

A. THREATS TO VALIDITY

1) CONSTRUCT VALIDITY

The construct validity of our approach is primarily influenced by our decision to focus on disclosures that name recipients, as opposed to those that merely categorize them. This decision stems from our preliminary manual inspection of privacy policies, where we observed a consistent lack of adherence to the detailed disclosure requirements set

forth in the GDPR guidelines. Specifically, none of the 100 policies we examined sufficiently detailed recipient categories (activity, sector, sub-sector, or location) in alignment with the official guidelines. Consequently, our automated method was calibrated to scrutinize disclosures that explicitly name recipients, a practice more in tune with the GDPR's transparency principle. While this approach enhances the relevance and specificity of our analysis, it introduces a limitation: the potential oversight of category-based disclosures that may, albeit infrequently, conform to GDPR standards. This exclusion could lead to an under-representation of compliant practices in our findings. Nevertheless, our decision to focus on named disclosures is justified by the higher likelihood of these practices aligning with GDPR transparency requirements, thereby reinforcing the construct validity of our study in capturing a critical aspect of GDPR compliance.

2) REPRODUCIBILITY AND REPEATABILITY

The ChatGPT API has demonstrated exemplary performance, positioning itself through this article as a robust alternative to conventional machine learning methods reliant on annotated privacy policies. The widespread appeal of this tool has culminated in substantial demand, leading OpenAI to impose request limitations and resulting in occasional server-side errors. The inherent non-deterministic nature of ChatGPT introduces variability in its outputs, potentially challenging result reproducibility. However, our observations indicate that newer versions of the GPT models (i.e., GPT-4 Turbo) appear to mitigate this output variability by providing a seed parameter, further affirming its suitability for extracting practices from privacy policies.

3) INTERNAL VALIDITY

Our proposed methods are statistical in nature, which can introduce the risks of false positives and false negatives when evaluating compliance with the GDPR transparency requirements. This poses a challenge to the accuracy of our claims. To address this, we've meticulously curated annotated datasets, allowing for the computation of validation metrics for each described component. Additionally, we validated each method using distinct data sets, sidestepping potential biases. During the crafting of prompts, we prioritized those leading to superior precision metrics, thereby reducing false positives and bolstering the reliability of our findings.

Our pursuit of libraries in the apps' execution traces has illuminated their significant role in personal data transfers, and their core role in the detected issues. Our search is bound by our libraries' white list, which we subsequently scout within the traces. Despite rigorous efforts to expand this list, we acknowledge that some libraries might escape our radar, leading to false negatives in our outcomes. Yet, utilizing a list ensures that identified entities are genuinely libraries. Furthermore, our approach relies on identifying connections made via the standard Android API. While most network connections are established through it, some others might be set up e.g., through native code, leading us to miss these data-sharing practices in Frida (although still intercepting them in the MitM proxy). These potential data transfers might yield false negative cases, yet they do not threaten the validity of our results in setting a lower threshold for undisclosed data-sharing practices.

4) EXTERNAL VALIDITY

For this study, we employed dynamic analysis tools to detect data transfers to third parties from the apps, including the Exerciser Monkey tool [51], which injects pseudo-random inputs into the apps, for the apps' stimulation during the dynamic analysis. As noted in related studies [52], the code coverage of such tools can be restricted, suggesting that many connections might not be triggered, leading to a potentially skewed representation of data transfers. However, our approach emphasizes soundness over completeness, ensuring that our observations and conclusions are truthful, albeit possibly incomplete. Thus, the actual situation might be even more concerning than our findings suggest, but never less severe than we exposed. Moreover, we relied on ChatGPT to check if two legal entities point at the same company and thus discern between the data controller and other recipients. This can be affected by the timeframe of the data used to train ChatGPT, as this can rapidly change in the business world. However, new GPT models like GPT-4 Turbo — Which is a modifiable parameter of our components —, have updated information up to April 2023, minimizing this problem.

VI. CONCLUSION

This paper has described a method to assess whether Android apps meet GDPR transparency requirements when transferring personal data. We applied it to 9,000 applications on the Google Play Store, yielding alarming results. An overwhelming 81.12% of applications fail to transparently declare the recipients of personal data in their privacy policies. This poses a significant risk to user privacy but also to app owners, who may face substantial financial penalties if do not meet GDPR transparency requirements. Furthermore, it also raises the question of the legal basis supporting the data collection by third parties, which in turn may challenge the lawfulness of these extended practices as already decided by the European Data Protection Board in a recent decision [53].

Our future work points to supporting developers to understand better whether they meet transparency requirements or gain awareness of the causes. Upon discovering that libraries are involved in almost three-quarters of the cases of noncompliance, we will investigate whether this may be due to a lack of transparency on the part of the responsible of these libraries, as previous work has shown [54]. We also aim to support data protection authorities in better spotting concerning issues that, due to their scale and impact, deserve their attention. All in all, our findings underscore the urgent need for more comprehensive and transparent data practices in app development and distribution markets.

AUTHOR CONTRIBUTIONS

David Rodriguez: Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft. **Jose M. Del Alamo:** Conceptualization, Investigation, Resources, Writing - Original Draft, Writing - Review & Editing, Supervision, Project administration, Funding acquisition. **Celia Fernández-Aller:** Conceptualization (legal assessment regarding privacy and GDPR requirements), Writing - Review & Editing. **Norman Sadeh:** Conceptualization, Writing - Review & Editing.

REFERENCES

[1] EUR-Lex. (2016). *EUR-Lex—32016R0679-EN*. [Online]. Available: https://eur-lex.europa.eu/eli/reg/2016/679/oj

[2] M. Goddard, "The EU general data protection regulation (GDPR): European regulation that has a global impact," *Int. J. Market Res.*, vol. 59, no. 6, pp. 703–705, Nov. 2017, doi: 10.2501/ijmr-2017-050.

[3] *The EU's Data Strategy from a Multifaceted Perspective. Views from Southern Europe*, PromethEUs, Amherst, NY, USA, Jun. 2023, pp. 1–76. [Online]. Available: https://www.i-com.it/wp-content/uploads/2023/06/PromethEUs_DataStrategy_Joint-Publication-Final-1.pdf

[4] C. J. Hoofnagle, B. van der Sloot, and F. Z. Borgesius, "The European union general data protection regulation: What it is and what it means," *Inf. Commun. Technol. Law*, vol. 28, no. 1, pp. 65–98, Jan. 2019, doi: 10.1080/13600834.2019.1573501.

[5] H. Wang, Y. Guo, Z. Ma, and X. Chen, "WuKong: A scalable and accurate two-phase approach to Android app clone detection," in *Proc. Int. Symp. Softw. Testing Anal.*, 2015, pp. 71–82, doi: 10.1145/2771783.2771795.

[6] European Commission. (2018). *Guidelines on Transparency Under Regulation 2016/679 (wp260rev.01)*. Accessed: Nov. 15, 2023. [Online]. Available: https://ec.europa.eu/newsroom/article29/items/622227

[7] European Data Protection Board. (2023). *Guidelines 01/2022 on Data Subject Rights—Right of Access*. Accessed: Nov. 15, 2023. [Online]. Available: https://edpb.europa.eu/our-work-tools/our-documents/guidelines/guidelines-012022-data-subject-rights-right-access_en

[8] Information Commissioner's Office. (2020). *Data Protection Act 2018. Enforcement Powers of the Information Commissioner*. Penalty Notice. TikTok Information Technologies U.K. Limited. [Online]. Available: https://ico.org.uk/media/4025182/tiktok-mpn.pdf

[9] Court of Justice of the European Union. (2023). *Judgment of the Court (First Chamber) of 12 January 2023. RW v Österreichische Post AG—Case C-154/21*. Accessed: Nov. 15, 2023. [Online]. Available: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A62021CJ0154

[10] J. M. Del Alamo, D. S. Guaman, B. García, and A. Diez, "A systematic mapping study on automated analysis of privacy policies," *Computing*, vol. 104, no. 9, pp. 2053–2076, Sep. 2022, doi: 10.1007/s00607-022-01076-3.

[11] P. Ferrara and F. Spoto, "Static analysis for GDPR compliance," in *Proc. Italian Conf. Cybersecur.*, vol. 2058, 2018, pp. 1–10.

[12] Q. Jia, L. Zhou, H. Li, R. Yang, S. Du, and H. Zhu, "Who leaks my privacy: Towards automatic and association detection with GDPR compliance," in *Wireless Algorithms, Systems, and Applications*. Cham, Switzerland: Springer, Jun. 2019, pp. 137–148, doi: 10.1007/978-3-030-23597-0_11.

[13] Z. Yu, S. Macbeth, K. Modi, and J. M. Pujol, "Tracking the trackers," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 121–132.

[14] T. Libert, "Exposing the invisible web: An analysis of third-party http requests on 1 million websites," *Int. J. Commun.*, vol. 9, pp. 3544–3561, Jan. 2015. [Online]. Available: https://ijoc.org/index.php/ijoc/article/view/3646

[15] S. Han, J. Jung, and D. Wetherall, "A study of third-party tracking by mobile apps in the wild," Dept. Comput. Sci. Eng., Univ. Washington, Seattle, WA, USA, Tech. Rep. UW-CSE-12-03, Jan. 2012.

[16] R. Binns, J. Zhao, M. V. Kleek, and N. Shadbolt, "Measuring third-party tracker power across web and mobile," *ACM Trans. Internet Technol.*, vol. 18, no. 4, pp. 1–22, Nov. 2018, doi: 10.1145/3176246.

[17] N. Vallina-Rodriguez, S. Sundaresan, A. Razaghpanah, R. Nithyanand, M. Allman, C. Kreibich, and P. Gill, "Tracking the trackers: Towards understanding the mobile advertising and tracking ecosystem," 2016, *arXiv:1609.07190*.

[18] D. Rodriguez, J. M. Del Alamo, M. Cozar, and B. García, "ROI: A method for identifying organizations receiving personal data," *Computing*, vol. 105, no. 12, Aug. 2023, doi: 10.1007/s00607-023-01209-2.

[19] W. Hu, D. Octeau, P. D. McDaniel, and P. Liu, "Duet: Library integrity verification for Android applications," in *Proc. ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jul. 2014, pp. 141–152, doi: 10.1145/2627393.2627404.

[20] L. Li, T. F. Bissyandé, J. Klein, and Y. Le Traon, "An investigation into the use of common libraries in Android apps," in *Proc. IEEE 23rd Int. Conf. Softw. Anal., Evol., Reengineering (SANER)*, vol. 1, Mar. 2016, pp. 403–414, doi: 10.1109/SANER.2016.52.

[21] M. Backes, S. Bugiel, and E. Derr, "Reliable third-party library detection in Android and its security applications," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: Association for Computing Machinery, Oct. 2016, pp. 356–367, doi: 10.1145/2976749.2978333.

[22] Z. Ma, H. Wang, Y. Guo, and X. Chen, "LibRadar: Fast and accurate detection of third-party libraries in Android apps," in *Proc. IEEE/ACM 38th Int. Conf. Softw. Eng. Companion (ICSE-C)*. New York, NY, USA: Association for Computing Machinery, May 2016, pp. 653–656.

[23] J. Zhang, A. R. Beresford, and S. A. Kollmann, "LibID: Reliable identification of obfuscated third-party Android libraries," in *Proc. 28th ACM SIGSOFT Int. Symp. Softw. Test. Anal.* New York, NY, USA: Association for Computing Machinery, Jul. 2019, pp. 55–65, doi: 10.1145/3293882.3330563.

[24] Y. He, X. Yang, B. Hu, and W. Wang, "Dynamic privacy leakage analysis of Android third-party libraries," *J. Inf. Secur. Appl.*, vol. 46, pp. 259–270, Jun. 2019, doi: 10.1016/j.jisa.2019.03.014.

[25] C. Schindler, M. Atas, T. Strametz, J. Feiner, and R. Hofer, "Privacy leak identification in third-party Android libraries," in *Proc. 7th Int. Conf. Mobile Secure Services (MobiSecServ)*, Feb. 2022, pp. 1–6, doi: 10.1109/MobiSecServ50855.2022.9727217.

[26] H. Cheng, G. Hu, J. Liu, Z. Kang, C. Pan, and Z. Zhang, "Detecting third-party libraries for privacy leakage in packed Android applications," in *Proc. China Autom. Congr. (CAC)*, Nov. 2022, pp. 5053–5058, doi: 10.1109/CAC57257.2022.10054907.

[27] V. Morel and R. Pardo, "SoK: Three facets of privacy policies," in *Proc. 19th Workshop Privacy Electron. Soc.* New York, NY, USA: Association for Computing Machinery, Nov. 2020, pp. 41–56, doi: 10.1145/3411497.3420216.

[28] M. V. Reiss, "Testing the reliability of ChatGPT for text annotation and classification: A cautionary remark," 2023, *arXiv:2304.11085*.

[29] P. Törnberg, "ChatGPT-4 outperforms experts and crowd workers in annotating political Twitter messages with zero-shot learning," 2023, *arXiv:2304.06588*.

[30] F. Gilardi, M. Alizadeh, and M. Kubli, "ChatGPT outperforms crowd-workers for text-annotation tasks," 2023, *arXiv:2303.15056*.

[31] J. H. Choi, K. E. Hickman, A. Monahan, and D. B. Schwarcz, "ChatGPT goes to law school," *J. Legal Educ.*, vol. 71, no. 3, p. 387, Jan. 2023, doi: 10.2139/ssrn.4335905.

[32] P. G. Kelley, L. F. Cranor, and N. Sadeh, "Privacy as part of the app decision-making process," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.* New York, NY, USA: Association for Computing Machinery, Apr. 2013, pp. 3393–3402, doi: 10.1145/2470654.2466466.

[33] Apple Developer. *App Privacy Details—App Store*. Accessed: Nov. 15, 2023. [Online]. Available: https://developer.apple.com/app-store/app-privacy-details/

[34] Google Play Store Support. (Mar. 2023). *Google Play's Data Safety Section*. [Online]. Available: https://support.google.com/googleplay/android-developer/answer/10787469?hl=en

[35] R. Khandelwal, A. Nayak, P. Chung, and K. Fawaz, "Unpacking privacy labels: A measurement and developer perspective on Google's data safety section," 2023, *arXiv:2306.08111*.

[36] D. Rodriguez, A. Jain, J. M. D. Alamo, and N. Sadeh, "Comparing privacy label disclosures of apps published in both the app store and Google Play Stores," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Jul. 2023, pp. 150–157, doi: 10.1109/EuroSPW59978.2023.00022.

[37] A. Jain, D. Rodriguez, J. M. D. Alamo, and N. Sadeh, "ATLAS: Automatically detecting discrepancies between privacy policies and privacy labels," in *Proc. IEEE Eur. Symp. Secur. Privacy Workshops (EuroS&PW)*, Jul. 2023, pp. 94–107, doi: 10.1109/eurospw59978.2023.00016.

[38] H. Zheng, M. Xue, H. Lu, S. Hao, H. Zhu, X. Liang, and K. Ross, "Smoke screener or straight shooter: Detecting elite Sybil attacks in user-review social networks," in *Proc. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, 2018, doi: 10.14722/ndss.2018.23009.

[39] Z. Tan and W. Song, "PTPDroid: Detecting violated user privacy disclosures to third-parties of Android apps," in *Proc. IEEE/ACM 45th Int. Conf. Softw. Eng. (ICSE)*, May 2023, pp. 473–485, doi: 10.1109/ICSE48619.2023.00050.
[40] 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 *Proc. 29th USENIX Secur. Symp. (USENIX Security)*, 2020, pp. 985–1002.
[41] D. S. Guamán, D. Rodriguez, J. M. del Alamo, and J. Such, "Automated GDPR compliance assessment for cross-border personal data transfers in Android applications," *Comput. Secur.*, vol. 130, Jul. 2023, Art. no. 103262. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404823001724
[42] I. Montani et al., Aug. 2023, "Explosion/spaCy: V3.6.1: Support for Pydantic v2, find-function CLI and more," *Zenodo*, doi: 10.5281/zenodo.8225292.
[43] Crunchbase. (2023). *Crunchbase*. [Online]. Available: https://www.crunchbase.com/
[44] Frida. (2023). *A World-Class Dynamic Instrumentation Framework*. [Online]. Available: https://frida.re/
[45] I. Square. (2019). *Okhttp*. [Online]. Available: https://square.github.io/okhttp/
[46] (Feb. 2023). *MyTracker Android SDK*. [Online]. Available: https://github.com/myTrackerSDK/mytracker-android
[47] Google Play Store. *Google Play SDK Index*. [Online]. Available: https://play.google.com/sdks/?hl=en-419
[48] (2023). *Maven Repository*. [Online]. Available: https://mvnrepository.com/
[49] M. Novović, "Privacy nutrition labels, app store and the GDPR: Unintended consequences?" *J. Data Protection Privacy*, vol. 5, no. 3, pp. 267–280, 2022.
[50] E. Commission. (Apr. 2018). *Article29—Transparency Guidelines*. [Online]. Available: https://ec.europa.eu/newsroom/article29/items/622227/en
[51] Google Android Developers. (Apr. 2023). *UI/Application Exerciser Monkey*. [Online]. Available: https://developer.android.com/studio/test/other-testing-tools/monkey
[52] P. Patel, G. Srinivasan, S. Rahaman, and I. Neamtiu, "On the effectiveness of random testing for Android: Or how I learned to stop worrying and love the monkey," in *Proc. IEEE/ACM 13th Int. Workshop Autom. Softw. Test (AST)*. New York, NY, USA: Association for Computing Machinery, May 2018, pp. 34–37.
[53] European Data Protection Board. (Dec. 2022). *Binding Decision 4/2022 on the Dispute Submitted by the Irish SA on Meta Platforms Ireland Limited and its Instagram Service (Art. 65 GDPR)*. [Online]. Available: https://edpb.europa.eu/system/files/2023-01/edpb_binding_decision_202204_ie_sa_meta_instagramservice_redacted_en.pdf
[54] K. Kollnig, R. Binns, P. Dewitte, M. Van Kleek, G. Wang, D. Omeiza, H. Webb, and N. Shadbolt, "A *Fait Accompli*? An empirical study into the absence of consent to third-party tracking in Android apps," in *Proc. 17th USENIX Conf. Usable Privacy Secur. (SOUPS)*. Berkeley, CA, USA: USENIX Association, 2021, pp. 181–195.

DAVID RODRIGUEZ received the B.S. degree in telecommunications engineering and the M.S. degree in cybersecurity from Universidad Politécnica de Madrid (UPM), Spain, in 2021 and 2022, respectively, where he is currently pursuing the Ph.D. degree.

From 2021 to 2023, he was a Research Assistant with the Telematic Systems Department. Since 2023, he has been a Graduate Teaching Assistant with the Telematic Systems Department, UPM. He holds an intelectual property registration and several papers in conferences and journals. His research interests include the development of static and dynamic automated tools for auditing mobile devices and the creation and integration of machine learning methods and large language models to assess legal compliance regulations in the mobile ecosystems.

Mr. Rodriguez has been awarded the Best Presentation Paper and the Second Best Presentation Paper in the International Workshop on Privacy Engineering, in 2022 and 2023, respectively.

JOSE M. DEL ALAMO received the Ph.D. degree in data privacy and protection from the Department of Telematic Systems Engineering, Universidad Politécnica de Madrid.

He has imparted various cybersecurity courses across degrees and master's programs and has extensively supervised student research. He is currently an Associate Professor (with tenure) with the Department of Telematic Systems Engineering, Universidad Politécnica de Madrid. He is the Principal Investigator of notable projects, such as Horizon Europe SUNRISE and Spanish Government funded autoGDPR, focusing on topics, such as critical infrastructure resilience and data protection assessment. He holds multiple patents in the field and has a significant presence in academic publications, and chairing the IEEE International Workshop on Privacy Engineering.

Dr. Alamo contributions have garnered several awards, such as the Young Scholar Award at the 8th International Conference on Computers, Privacy, and Data Protection. He received multiple honors, including two Research Sexenniums.

CELIA FERNÁNDEZ-ALLER received the Ph.D. degree in law and technology.

She is currently a Senior Lecturer with Universidad Politécnica de Madrid, where she manages courses on the legal and ethical aspects of computer science. She was previously a Lecturer with Universidad Centroamericana de El Salvador (UCA) and had an internship with Comillas Pontifical University. She is an Active Member with the GIOS Research Group and was appointed by the Spanish Government to draft a digital rights charter. She has been honored as a Visiting Professor at Bristol University and serves on several advisory boards. She has contributed to conferences, such as ECAI and ECSA, and has published extensively in journals, such as DOXA and *IEEE Technology and Society Magazine*. She holds two patents in EDUCERE Project and has issued various legal reports for companies and public institutions. Her research interests include transdisciplinary studies on human rights, data protection, and emerging technologies, such as AI and robotics.

NORMAN SADEH (Member, IEEE) is currently a Professor with the School of Computer Science, Carnegie Mellon University (CMU), and a leading expert in fields ranging from cybersecurity and online privacy to artificial intelligence and supply chain management. He directs CMU's Mobile Commerce and E-Supply Chain Management Laboratories and Co-Founded its Ph.D. Program in societal computing. Notably, he was the Founding CEO of Wombat Security Technologies, which was acquired by Proofpoint, in 2018. The technologies from this venture protect tens of millions of employees globally. He has influenced product designs at tech giants, such as Apple, Google, and Facebook. He has served in key roles shaping European cybersecurity and privacy research and has authored over 300 scientific publications.

He was honored with the 2018 Outstanding Entrepreneur of the Year Award from the Pittsburgh Venture Capital Association.