The Effect of Platform Policies on App Privacy Compliance: A Study of Child-Directed Apps

Noura Alomar
University of California, Berkeley
Berkeley, California, USA
nnalomar@berkeley.edu

Joel Reardon
University of Calgary
Calgary, Alberta, Canada
joel.reardon@ucalgary.ca

Aniketh Girish
IMDEA Networks Institute
Madrid, Spain
aniketh.girish@imdea.org

Narseo Vallina-Rodriguez
IMDEA Networks Institute
Madrid, Spain
narseo.vallina@imdea.org

Serge Egelman
ICSI
University of California, Berkeley
Berkeley, California, USA
egelman@cs.berkeley.edu

Abstract

Over the past few years, the two dominant app platforms made major improvements to their policies surrounding child-directed apps. While prior work repeatedly demonstrated that privacy issues were prevalent in child-directed apps, it is unclear whether platform policies can lead child-directed apps to comply with privacy requirements, when laws alone have not. To understand the effect of recent changes in platform policies (e.g., whether they result in greater levels of compliance with applicable privacy laws), we conducted a large-scale measurement study of the privacy behaviors of 7,377 child-directed Android apps, as well as a follow-up survey with some of their developers. We observed a drastic decrease in the number of apps that transmitted personal data without verifiable parental consent and an increase in the number of apps that encrypted their transmissions using TLS. However, improper use of third-party SDKs still led to privacy issues (e.g., inaccurate disclosures in apps' privacy labels). Our analysis of apps' privacy practices over a period of a few months in 2023 and a comparison of our results with those observed a few years ago demonstrate gradual improvements in apps' privacy practices over time. We discuss how app platforms can further improve their policies and emphasize the role of enforcement in making such policies effective.

Keywords

Software developers, SDKs, privacy, security, Google Play, personal data, development practices, Android, COPPA

1 Introduction

Regulations in the United States [34, 102] and Europe [79] require that child-directed online services provide heightened privacy protections for their users. Investigating the effects these regulations have had on the behaviors of child-directed online services has been the subject of several prior studies (e.g., [15, 32, 33, 58, 59, 90, 94, 114, 138]). For example, in 2018, Reyes et al. [116] conducted a large-scale analysis of the network traffic generated from nearly six thousand child-directed mobile apps and showed that the data collection and sharing behaviors of 57% of them rendered them potentially in violation of the Children's Online Privacy Protection Act (COPPA), a US law that has existed for over 25 years. The identified violations were due to incorrect data transmission, consent handling, disclosure, and software configuration practices of the general developer population [18, 37, 57, 101, 129, 133, 136]. These findings have motivated regulators, such as the Federal Trade Commission (FTC) [64], to take action; this included new rulemaking efforts [35], as well as enforcement actions (e.g., [65–67]).

Industry has taken notice: both major mobile platforms adopted new policies to help app developers comply with COPPA and other applicable privacy laws [132]. This paper examines how Google has changed its policies to improve privacy amongst child-directed Android apps [21, 26], and whether this resulted in greater levels of compliance with applicable privacy regulations. For example, all developers are now required to prepare privacy labels—"data safety labels" in Google's parlance [22]—that disclose their apps' data handling practices, developers of child-directed apps can now only embed third-party advertising Software Development Kits (SDKs) approved by Google [77], and cannot collect the Android Advertising ID (AAID) from children [21]. Research has shown that developers' understanding of their compliance obligations under applicable privacy regulations is influenced by the stringency of platform policy enforcement efforts [6]. While Google has made various changes to its developer policies over the past five years [26], to our knowledge, the real-world impact of these changes on the privacy behaviors of child-directed apps has not previously been studied. Understanding the effects of policy changes on the privacy practices of child-directed apps is essential for shaping future policy efforts aimed at regulating the practices of other types of apps.

We conducted a systematic large-scale analysis of the extent to which child-directed Android apps were potentially in violation of the Google Play Store policies [21]. We specifically focused on measuring the prevalence of privacy issues that would render developers in violation of these policies, and which can also be considered potential violations of privacy regulations (e.g., COPPA). We measured the prevalence of privacy issues related to: (1) collecting or sharing personal data from children, (2) making inaccurate

Proceedings on Privacy Enhancing Technologies 2025(3), 170–191

https://doi.org/10.56553/popets-2025-0094

disclosures in apps' privacy labels [22], and (3) using privacy configurations of third-party SDKs embedded in child-directed apps in potentially-violative ways (e.g., that result in illegal user profiling and/or behavioral advertising). We also supplement our technical analyses with the results of a follow-up developer survey to gather additional explanatory data. We answer the following research questions:

RQ 1 Have the updates made to Google Play's policies resulted in improved compliance rates among child-directed apps?

RQ 2 How prevalent are third-party SDK privacy misconfigurations and data safety label disclosure mistakes among child-directed apps?

RQ 3 Are developers giving sufficient consideration to third-party SDK privacy configurations and Google Play's data safety labels in their development processes?

Through our analysis of 7,377 child-directed apps tested from early February to early July of 2023 and a comparison of our results with those of Reyes et al. [116], we observe an overall improvement in the privacy practices of child-directed apps. For example, we show that the number of apps that transmitted AAIDs decreased from 59% to 8.8% and those that collected geolocation coordinates decreased from 3% to 0.1% [116]. However, we also find that developers were still struggling with understanding the behaviors of third-party SDKs, which led many of them to be unable to prepare accurate privacy labels for their apps. While the adoption of some of third-party SDKs' privacy configurations has increased, there were child-directed apps that were still using third-party advertising SDKs that are not allowed by Google Play's policies [73, 77], as well as incorrectly configuring those that are (i.e., included on Google's list of self-certified SDKs [77]). The results of our survey support these findings: 60% did not use technical testing tools to identify the data types that need to be disclosed in their privacy labels; while 74% of respondents were familiar with SDKs' privacy configurations, 64% stated that using them was challenging. While new platform policies have largely been effective at increasing compliance amongst child-directed apps, more work needs to be done to educate developers about the responsible (and legal) use of third-party SDKs *vis-à-vis* relevant privacy regulations.

2 Background

This section describes the Google Play policies that are applicable to child-directed apps and briefly summarizes the kids provisions of COPPA, GDPR, and the CCPA. The latter is included to demonstrate how the privacy issues that we investigated would be considered potential violations of platform policies and privacy laws across varying jurisdictions.

2.1 COPPA, GDPR, and CCPA

The Children's Online Privacy Protection Act (COPPA) [34] regulates the practices of online services used by users under the age of 13 in the US. It applies to services that have actual knowledge of their use by child users or use any type of content that can attract them [36]. Depending on the purpose of collecting personal data, COPPA might require obtaining verifiable parental consent prior to doing so (e.g., opting in to behavioral advertising) [34, 36]. Services operating in the US targeting child users in California are additionally required to comply with the kids provisions of the California Consumer Privacy Act (CCPA) [102]. The sale of personal information collected from those under 13 is considered a violation of these provisions if parental consent is not obtained [102].

Similarly, the General Data Protection Regulation (GDPR) [79] has provisions that aim to protect the privacy of users below 16 years of age in the EU. According to Article 6 of the GDPR, online services collecting personal data from users in the EU are required to have one of the six legal bases for data processing, which include fulfilling legal obligations, serving legitimate interests and having user consent [79]. For services that are targeted at child users, the GDPR has additional provisions in Article 8 that state that user consent cannot be relied upon as a legal basis unless services are able to verify that they obtained it from guardians [79].

These privacy laws have common disclosure and consent requirements that apply to child-directed apps, which are also reflected in Google Play's families policies [21]. For instance, obtaining affirmative parental consent before transmitting personal data to third parties for certain purposes is required under COPPA [34], can be used as a basis for data processing under the GDPR [79], and can justify selling children's data to third parties under the CCPA [102].

2.2 Google Play Store Policies

Publishing apps on Google Play is subject to security and privacy policies [26, 109], which have been in alignment with privacy law requirements. While research has shown that they are insufficiently enforced [6, 116], developers cannot submit their apps to the store without: (1) certifying that they are compliant, (2) indicating their apps' target audiences, (3) providing links to their privacy policies, and (4) disclosing whether they use advertisements [74, 75, 108]. Google Play extended its policies in 2022 to also require making certain disclosures about apps' data practices in the form of privacy labels [22]. While it is unclear whether Google Play checks the accuracy of these disclosures, making inaccurate disclosures is still considered a "deceptive" practice [24, 28].

Google Play also requires developers to disclose their apps' data collection and sharing behaviors in their privacy policies, ensure that users' explicit consent is obtained before transmitting their personal data (for certain uses), and avoid insecure communications [20, 72]. The policies also regulate the use of persistent and resettable identifiers that allow tracking of users (e.g., AAIDs, BSSIDs and IMEIs) [71, 73]. For advertising, developers can only use the AAID, which can be reset using system settings [71]. For it to preserve this property, Google Play prohibits "linking" it to non-resettable identifiers and requires obtaining consent before transmitting it with other types of personal data [19, 20, 23].

Google Play's Designed for Families (DFF) program introduced additional policies that apply specifically to child-directed apps [21, 73, 75]. They prohibit collecting certain types of personal data (e.g., AAIDs and geolocation data) and protect access to them through Android permissions (e.g., the `AD_ID` and `ACCESS_FINE_LOCATION` permissions) [21, 73]. Alternatively, developers can use the "app set ID" (Section 5.3.2) for permitted purposes (e.g., analytics), which has better privacy properties compared to existing identifiers (e.g.,

AAIDs or IMEIs) [71, 73]. Accordingly, the families policies prohibit transmitting app set IDs alongside AAIDs (or other persistent identifiers) or using them to enable behavioral targeting [23].

The policies also restrict the use of third-party advertising SDKs in child-directed apps [25, 27, 73]. Apps whose primary audience is children are only allowed to embed specific SDKs and versions listed in Google Play's list of self-certified providers (e.g., not older than `4.0.1` for Unity Ads [54]) [77]. The SDKs included in this list ostensibly either do not perform behavioral advertising or user profiling, or include configurations that allow disabling that functionality for children. Accordingly, developers are required to correctly configure these SDKs for child-directed treatment [25, 27, 73, 77] (Section 4.3). For example, in apps that target Android versions 12 and below, where the `AD_ID` permission does not exist, use of these configurations is necessary to prevent collecting AAIDs by these third parties [44, 46].

3 Related Work

Prior work has examined how mobile app developers make various security and privacy mistakes, including incorrect use of encryption algorithms [55, 57, 105, 116], not disclosing data collection practices [70, 88, 91, 104, 133, 134], data sharing without user consent [87, 89, 90, 100, 114, 116], over-privileging apps with permissions [60], enabling data exfiltration through side channels [115] and delaying applying security updates [50, 137].

Several studies measured the prevalence of these mistakes in child-directed apps over the past few years [90, 114, 116, 126]. An analysis of network transmissions collected from users of more than 14,000 apps by Razaghpanah et al. [114] showed that 24% of those that targeted children shared personal data with trackers. Reyes et al. [116] quantified potential COPPA violations in around 6,000 DFF-approved apps in 2018 by relying on a testing pipeline that used input generated by the Android Application Exerciser Monkey [41]. They identified privacy issues in 57% of these apps, such as transmitting personal data without parental consent, linking of device identifiers and not using TLS [116]. Subsequent investigations by Kollnig et al. [90] and Sun et al. [126] confirmed that the same privacy issues continued to exist in child-directed apps.

User consent has also been consistently found to not be appropriately collected or handled by the majority of apps that target general audiences [87, 89, 100, 101]. In separate studies, Koch et al. [87] and Kollnig et al. [89] showed that only 22% and 10% of the apps they analyzed, respectively, implemented user consent mechanisms. A dynamic analysis of more than 86,000 apps by Nguyen et al. [100] also showed that 34% defaulted to sharing personal data with advertisers without obtaining user consent. Shortcomings in the implementation of consent mechanisms were also identified, such as transmitting personal data before users respond to consent prompts or after they decline data collection [87, 101].

To address these security and privacy issues, the Google Play Store [26] and the Apple App Store [14] introduced various changes to their policies in recent years. One of which requires developers to disclose their practices as succinct privacy labels, which have been found to suffer from inaccuracies [86, 88, 91, 134]. This is partly caused by developers experiencing difficulty in aligning the defined disclosure specifications for privacy labels with their actual data practices [86, 92]. The impact of the remaining policy changes on developer practices is yet to be understood. In this paper, we build upon that of Reyes et al. [116] to uncover how the platform policies introduced since 2018 impacted the practices of child-directed apps. Our investigation is based on measurements of various privacy issues, which include improper transmissions of personal data, not using TLS, and not having complete disclosures in apps' privacy labels. Our work also complements works that studied developer adoption of SDK privacy configurations (e.g., [6, 98]) to further understand the extent to which child-directed apps configure third-party SDKs for compliance with Google's policies and the requirements of applicable privacy regulations.

4 Methods

This section explains how we identified and tested a large corpus of child-directed apps, as well as the heuristics we used to quantify the prevalence of potential policy violations in these apps.

4.1 Identification of Child-Directed Apps

We scraped the Google Play Store to identify apps that: (1) were advertised as compliant with the Google Play Store Families program or Google Play's Teacher Approved program (i.e., their app listings included the statement "Committed to follow the Play Families Policy" [21][1] or the "Teacher Approved" badge [107]); or (2) had titles or descriptions that included keywords that signified that children are among their target audiences. To do so, we identified a set of keywords that we expected to help us find child-directed apps on the store and then searched for them in apps' titles and descriptions. These are: "kid," "baby," "babies," "preschool," "school," "ABC," "kindergarten," "first grade," "second grade," "third grade," "fourth grade," "fifth grade," "coloring," "learn," "spelling," "child," "children," "toddler," "alphabet," and "math." We chose them based on an initial analysis of the titles and descriptions of a subset of child-directed apps that we identified through manual searches.

Using this process, we identified 11,490 apps that appeared to be child directed. Since our keyword searches could have resulted in false positives, as a sanity check, we manually examined the results and excluded apps that did not appear to be child-directed. This review allowed further confirming that the apps are child-directed based on information communicated in their titles, descriptions, Google's badges or screenshots that featured characters that were likely to be attractive to children. We then crawled the store over two different periods of time in 2023 to collect two complementary datasets containing the identified child-directed apps that we queued for testing and their associated metadata (Section 4.2).

4.2 Testing of Child-Directed Apps

We used dynamic analysis methods established in the literature to capture network traffic data and decode transmissions exchanged between the tested apps and remote end-points [6, 68–70, 93, 116, 117]. Each app was run for 10 minutes on a custom version of Android and fed auto-generated input by Android's Application Exerciser Monkey [41]. To obtain visibility into encrypted traffic, we instrumented the functions that apps used to read or write to TLS sockets, which allowed observing data before encryption and

[1] We consider these enrolled in Google Play's Designed for Families (DFF) program.

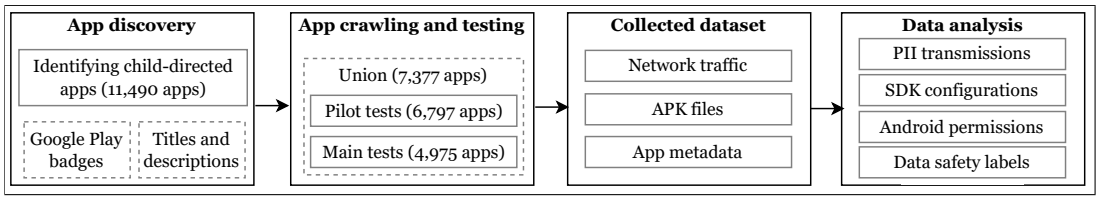

Figure 1: Overview of our data collection approach.

Table 1: Breakdown of the apps tested during the two testing periods.

	First dataset	Second dataset	Union of two datasets
# of tested child-directed apps	6,797	4,975	7,377
# of DFF-badged apps [21]	3,095 (45.5%)	3,085 (62.0%)	3,684 (49.9%)
# of teacher-approved apps [107]	1,334 (19.6%)	983 (19.8%)	1,590 (21.6%)
# of apps whose titles indicated that they are targeting kids	2,520 (37.1%)	1,940 (39.0%)	2,776 (37.6%)
# of apps whose descriptions indicated that they are targeting kids	6,044 (89.0%)	4,372 (87.9%)	6,521 (88.4%)

after decryption. We then searched for encodings and permutations of personal information (e.g., AAIDs, Android IDs/SSAIDs, app set IDs, Firebase installation identifiers [FIDs], and geolocation data) in the network traffic that we observed in plaintext. We also used regular expressions to find privacy signals in communications exchanged with third-party SDK servers (Section 4.3). To perform static analysis, we relied on Android tools [45], such as the Android Debug Bridge (ADB) [42] and the Android Asset Packaging Tool (AAPT) [39]), to identify the Android versions targeted by the apps and the permissions they requested.

As a pilot, we queued the identified apps for dynamic testing from early February to early March of 2023 using Google Pixel 3a phones running Android 12 in California. We then worked on improving our instrumentation to fix some issues that we detected in the pilot study, and added support to detect the app set ID [40] and additional types of personal data. We then used that improved instrumentation for a second round of testing. Of the apps that we initially identified as being child directed, many apps could not be installed during testing. We determined that this was due to a combination of factors: some apps were not available in our region, some were paid, some had been removed from the store during the period between identifying child-directed apps by scraping the Play Store website and queueing them for testing, and others were not compiled for our particular hardware configuration.[2]

From mid-May to early July of 2023, we re-queued all of the previous apps for testing and added 302 additional apps that we subsequently identified using the same methodology. However, in addition to many apps not being available to download for the reasons previously described, we also found that not all of the apps that we initially tested were still available. This resulted in reducing the number of apps that we were able to successfully test in the later round of testing. For this reason, we decided to also report on the pilot results, when appropriate. Table 1 provides more details on the number of apps successfully tested in both rounds of testing.

We tested 7,377 unique apps released by 3,390 unique developers, cumulatively installed by 11.5 billion devices at the time of testing. Sixty-eight percent of the 7,377 apps were categorized as educational apps by the store. After comparing the two datasets, we found that 4,395 apps were tested in both testing rounds, 2,402 were tested only in the initial pilot, and 580 apps were only tested in the second testing round. We revisited the apps that we were not able to install in our second round of testing and found that this could have resulted from Google enforcing a new policy that prevented apps targeting Android versions older than 11 (API levels below 30) from being installed on devices running more recent Android versions [17, 29, 49, 76]. The difference between the number of apps successfully tested in the two runs could also be explained by the existence of apps that were not available on the store at the time of testing, or apps that their developers converted from free to paid or made them no longer available for use from California. Figure 10 provides a timeline showing when our datasets were collected and when relevant policies were introduced (Appendix G). Our results can be replicated using publicly-available data about the apps that were available during the same periods of time in 2023 (e.g., by downloading apps from archive sites such as APKPure [9]).

In Section 5, we report on all 7,377 unique apps tested, while considering the most recent versions of apps that we tested in both testing rounds (when two different versions of the same app were tested). We also indicate when certain analyses apply to only one dataset (e.g., developers' adoption of the app set ID, which we only measured during the second round). Whenever possible, we compare our results to those reported by Reyes et al. [116] to understand how the Google Play policies that were introduced since 2018 impacted the privacy practices of child-directed apps. We chose Reyes et al. [116] as a basis for our temporal comparisons since we targeted the same population of apps and used similar testing methods. Figure 1 summarizes our data collection approach.

[2]While Android 13 was the most recent version of Android available at the time of testing, we wanted to examine a prior version that did not regulate access to the AAID via the permissions system, allowing us to examine whether SDK privacy settings were being correctly configured by app developers. This still allowed us to examine whether apps targeting Android 13 were using the new `AD_ID` permission by performing static analysis on the `AndroidManifest.xml` file. More importantly, at the time of our testing, 85% of Android users were using a version older than 13 [125], and therefore testing Android 13 would not necessarily be representative.

4.3 Testing of Privacy Configurations

Developers of child-directed apps are required to configure third-party SDKs for privacy compliance [25, 77]. Privacy configurations offered by third-party SDKs can help developers signal to data recipients that: (1) personal data was collected from a child user, (2) it was collected from a region where the provisions of a specific privacy regulation apply, or that (3) a user has not consented to using their data for certain purposes. Chartboost [51], for example, provides a client-side method, `addDataUseConsent`, which can help developers comply with COPPA, GDPR, or the CCPA by setting three corresponding privacy flags (`coppa`, `pidatauseconsent` and `us_privacy`) in outbound transmissions.[3] Figure 2 provides an example of a server-side configuration provided by ironSource [82].

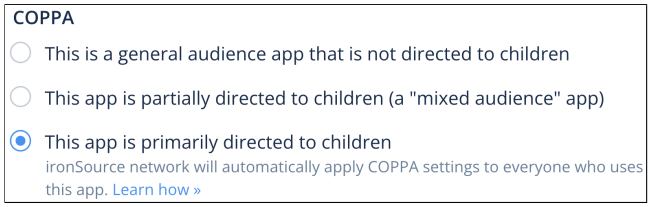

Figure 2: ironSource's [82] server-side COPPA control.

We tested the privacy configurations offered by a number of third-party SDKs that were either on Google's list of self-certified advertising SDKs [77] or were found to be used in child-directed apps in prior work (e.g., Meta's SDKs [99]) [116]. We examined whether their terms of service allowed integrating them in child-directed apps, and for those that did, we identified the privacy controls that their documentation instructed developers to use. We then integrated these SDKs in prototype Android apps we created and used each of the identified privacy configurations to observe how their usage manifested in network traffic. This allowed us to: (1) understand how SDK providers expect their SDKs to be configured in child-directed apps, (2) extract the privacy flags that get included in inbound[4] or outbound transmissions as a result of using privacy configurations, (3) build an understanding of how each possible flag is set, (4) identify the correct values that developers are supposed to use, and (5) build a corpus of API endpoints corresponding to all of the tested SDKs. We also investigated the default values of each identified flag and the specific configurations that would prevent collecting certain types of data (e.g., AAIDs).

We then searched for the identified privacy flags within the captured network traffic of the tested apps. We complemented our traffic searches with Frida-based [113] instrumentation to monitor the invocation of client-side SDK privacy methods and their parameters by apps. For example, developers of apps using ironSource's advertising SDK [82] can use the `setMetaData` method to set a number of privacy flags (e.g., correctly set `is_child_directed` to `true`). Our approach enabled detecting when privacy configurations were used by different third-party SDKs integrated in the apps we tested and not necessarily called by first-party code.

We also explored whether the design of privacy configurations impacted developers' level of adoption and enabled transmissions of children's personal data (Section 5.3.1). For example, we found SDKs that transmitted AAIDs to their servers, even when privacy configurations were used correctly. While the personal data included in these transmissions were flagged as collected from children, which signals to recipients that they should not be used for behavioral advertising, the mere collection of unused personal data may be at odds with data minimization requirements, such as under GDPR [103], as well as Google Play's policies [73].

4.4 Analysis of Data Safety Labels

Since July 2022, Google Play has required disclosing apps' privacy practices in standardized privacy labels [22]. To do so, developers fill out a questionnaire that asks them to provide details about their apps' data transmission and handling practices before they submit their apps to the store [22, 108]. Using the questionnaire, they can choose from a set of pre-defined data types (e.g., "device IDs or other IDs," "location," and "personal info") and purposes of data collection or sharing (e.g., "app functionality" and "advertising or marketing"), which then get disclosed as part of their app listings [22]. They are also expected to follow a specific criteria defined by Google Play to indicate whether each disclosed data type is *collected* or *shared*,[5] and whether encryption is used for communication security [22]. Examples are provided in Appendix F.

We evaluated whether developers were able to prepare accurate privacy labels. To do so, the labels for the apps in our corpus were scraped at the time of testing. We then inspected them to understand whether the personal data that we observed being collected or shared were disclosed. For that, we identified the domain names that we observed receiving personal information from the tested apps to identify the recipient, and then reasoned about the data collection purpose by examining their website, API documentation, and contents of the transmission. For apps that did not use TLS in their transmissions that contained personal data, we also examined whether doing so was accurately disclosed in their labels.

We followed three heuristics to obtain a lower bound for the number of apps that had disclosure mistakes in their privacy labels. First, we looked for whether each transmitted data type was disclosed, regardless of whether the data types were disclosed as *collected* or *shared*. Second, we considered "analytics", "personalization" and "advertising" as similar purposes because a number of third-party recipients' policies state they will use received data for more than one of these purposes (e.g., Unity Ads [53, 127], ironSource [83, 84], Start.io [123] and Meta [96, 99]). Third, we looked for whether a purpose of data collection and/or sharing was disclosed at all, regardless of whether it was disclosed for the specific data types that we observed being transmitted.

5 Results

In this section, we measure the prevalence of privacy issues across child-directed apps. Overall, we observed a decrease in the number of apps that transmitted personal data to first- or third-party

[3] According to Chartboost's developer documentation [51], developers can disable targeted advertising by setting the values of `coppa` to "`true`," `pidatauseconsent` to "`0`," or `us_privacy` to "`1NY-`."

[4] In some cases, inbound transmissions reveal server-side privacy configurations.

[5] Google Play's specification for data safety labels details the circumstances under which a data transmission should be disclosed as *collection* or *sharing* (e.g., data types transmitted to analytics services should be disclosed as *collected* data types since such services are considered service providers) [22].

recipients or did not use TLS. However, we also show that use of third-party SDKs is still contributing to other privacy issues.

5.1 Access and Collection of Personal Data

We observed 828 (11.2%) of the 7,377 apps transmit AAIDs, geolocation data, WiFi MAC addresses, router MAC addresses, router SSIDs, names, or email addresses to first- or third-party recipients. Of these apps transmitting personal data, 44% (363 apps) were DFF-badged and 8% were teacher-approved. Across the two testing periods, 172 apps (21% of 828 observed transmitting personal data) released new versions that we observed no longer transmitting certain types of personal data during the second testing round. Of the 657 apps that continued to transmit personal data, 39% were DFF-badged and 8.5% were teacher-approved. The transmissions that contained personal data were either initiated by the tested apps or system processes, such as Google Mobile Services (GMS) [8] and Google Services Framework (GSF), which can be called by apps to request certain functionality. We also observed GMS and GSF transmit AAIDs, SSAIDs, and FIDs [47] to `app-measurement.com` alongside app package names,[6] which allows Google to gather analytics data at scale. Additionally, 368 apps transmitted FIDs alongside AAIDs to `app-measurement.com`, which allows linking these two identifiers (and "bridging" AAID resets, negating the system privacy controls). Of these 368 apps, 234 (63.6%) were advertised as compliant with DFF; for 284 of these apps (77%), Google was the only recipient of personal data.

Table 2 provides details on the number of apps that transmitted each type of personal data captured by our instrumentation. It also illustrates how these numbers change after excluding the apps that ceased transmitting the same types of personal data in their subsequent versions. Figure 3 shows the top 20 recipients of personal data (excluding SSAIDs, app set IDs, and FIDs)[7] across the 7,377 apps. Other less common recipients include Umeng [128] (8 apps), OneSignal [106] (7 apps), Singular [120] (7 apps), Applifier (6 apps), and Kochava [119] (4 apps).

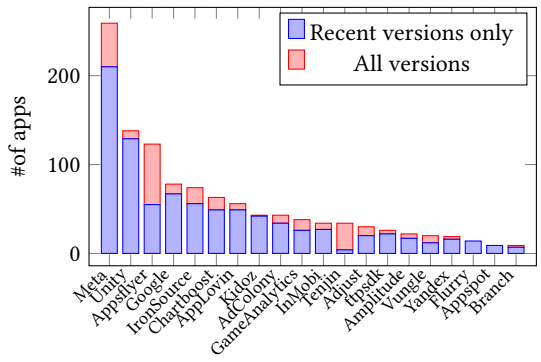

Figure 3: Top 20 data recipients across all app tests.

In the following sections, we summarize our results for each type of personal data and compare our findings with those of Reyes et al. [116], to directly compare how COPPA compliance has likely changed over the intervening years.

[6]Each Android app is uniquely identified by its package name, which also allows it to be located in the Google Play Store.

[7]We excluded these identifiers because they are unique to each app installation, and therefore do not directly allow for users to be tracked across apps and services.

5.1.1 Identifiers. Google Play lists various identifiers that child-directed apps are not allowed to collect [73]. Our results show that many child-directed apps collected AAIDs and two collected WiFi MAC addresses, both of which are prohibited because they allow long-term tracking of child users (see Table 2).

AAIDs. More than 11% of the 7,377 apps transmitted AAIDs, 172 of which released updated versions that stopped transmitting AAIDs in the second round of testing. Compared to the analysis conducted on 5,855 DFF apps by Reyes et al. [116], our results show a noticeable decrease in apps that transmitted AAIDs. While 59% of the apps analyzed by Reyes et al. [116] did so, our results show that this percentage decreased to 8.8% (652 apps).

In recent versions of Android (13 or later), AAIDs cannot be accessed unless the `AD_ID` permission is declared [71, 118]. To examine whether apps are able to use AAIDs, we statically analyzed apps' `APK` files to identify apps that either targeted older Android versions (i.e., `targetSdkVersion` below 33) or declared the `AD_ID` permission when targeting recent Android versions. Of the 7,377 apps, 1,460 (20%) targeted Android 13 or 14, whereas the rest targeted older versions that allow default AAID access. Furthermore, 578 of the 1,460 apps did not target Android 13 or 14 in our pilot tests, but updated their apps to do so in our recent tests. Similarly, 341 of the apps that targeted Android versions below 33 updated to Android 12 (i.e., `targetSdkVersion` 31 or 32) and 16 apps downgraded from Android 13 to a lower version, therefore they were still able to access AAIDs without declaring the permission.

Of the 1,460 apps, 617 (42%) declared the `AD_ID` permission. However, the permission was also declared in 1,284 of the apps that targeted Android versions older than 13 (i.e., the permission is not needed for accessing AAIDs in these versions). This could be explained by developers mistakenly declaring the permission in their apps when uncritically following third-party SDK integration instructions. We also found 257 apps that added the `AD_ID` permission in their recent versions and 124 apps that updated their apps to remove it. Of the 652 apps that transmitted AAIDs (see Table 2), only 84 apps (13%) needed to use the `AD_ID` permission because they targeted Android 13, whereas the rest (568 apps) targeted older Android versions. Of the 1,901 apps that declared the `AD_ID` permission, 70% were DFF-badged at the time of testing.

WiFi MAC Addresses. Only 2 apps transmitted WiFi MAC addresses, one of which did not do so in its updated version. Both of them were not DFF-badged. One of them shared AAIDs, SSAIDs and WiFi MAC addresses with Umeng [128], enabling tracking of children by linking different AAIDs that belong to the same device or SSAIDs associated with different app developers.

SSAIDs. Android IDs were transmitted by 720 apps, 109 of which stopped doing so in their recent versions. While SSAIDs cannot be relied upon for tracking users across apps released by different developers since the release of Android 8 in 2017, they can still allow tracking, as their values do not get reset after apps get uninstalled or updated, but rather only due to a factory reset of the device [38]. Additionally, transmitting SSAIDs alongside certain types of personal data to the same recipients is considered a violation of Google's policies [19]. Across all the apps we tested, SSAIDs were transmitted

alongside AAIDs (168 apps), FIDs (26 apps), WiFi MAC addresses (2 apps), and router SSIDs (1 app) to first- and third-party recipients such as Unity [127], Chartboost [31], and Yandex [135]. In total, 191 apps included at least one type of personal data with SSAIDs in the same transmissions to domains other than `app-measurement.com`.

AAIDs were transmitted alongside FIDs (13 apps), fine location coordinates (3 apps), WiFi MAC addresses (1 app), router BSSID (1 app) or IP location (1 app) to the same recipients. These included Amplitude [7], AppsFlyer [13], Yandex [135], and Umeng [128]. Thus, the total number of apps that transmitted AAIDs alongside location data (including router MAC addresses), FIDs, SSAIDs, or WiFi MAC addresses to domains other than `app-measurement.com` is 179. Sharing these data types alongside AAIDs allows converting resettable AAIDs to persistent IDs that can be used for long-term tracking, as it allows recipients to link AAIDs that belong to the same device after they have been reset. However, Reyes et al. [116] showed that more than 66% of the 5,855 apps they tested shared a persistent identifier (e.g., IMEIs or SSAIDs in Android versions that are older than 8) alongside AAIDs. This shows a drastic decrease in the number of apps that were attempting to link AAIDs to other types of personal data (RQ1), particularly those that might persist over time (e.g., router MAC addresses or fine location coordinates).

5.1.2 Location Data. Seven apps transmitted geolocation coordinates, three of which transmitted approximate geolocation coordinates only, whereas the rest transmitted exact geolocation coordinates. Additionally, we identified another 34 apps that transmitted or received IP location (e.g., using IP geolocation services such as ipwhois.io [80]), but we did not consider them as cases of data leakage since relying on IP addresses for obtaining user locations does not often lead to finding accurate user locations. However, 194 apps declared at least one of Android's location permissions (see Table 4 in Appendix B). Thus, even though only 7 apps transmitted fine or coarse location coordinates, the number of apps that had the capability of accessing children's location data exceeded that number. We also found 160 apps that potentially violated Google Play's policies by declaring the `ACCESS_FINE_LOCATION` permission, which allows accessing child users' exact locations [21].

However, our results also suggest a considerable decrease in the number of apps that have the capability of accessing location data. Only 2.6% of the apps we tested declared `ACCESS_FINE_LOCATION` or `ACCESS_COARSE_LOCATION`, whereas 12% of the apps tested by Reyes et al. [116] declared one of these permissions. Additionally, only 9 apps collected coarse or fine geolocation coordinates or seemed to have the capability of knowing users' approximate or exact locations using router MAC addresses or router SSIDs, whereas Reyes et al. [116] identified 256 apps that collected these types of data (see Table 2).

5.1.3 Contact Information. We identified only one app that collected names and email addresses. The app was DFF-badged and transmitted both data types to (`www.googleapis.com`). No apps in our corpus transmitted phone numbers. We investigated the number of apps that declared `GET_ACCOUNTS` or `READ_PHONE_STATE`, which are the permissions that regulate access to names, email addresses and phone numbers. We found 605 and 116 apps that declared `READ_PHONE_STATE` and `GET_ACCOUNTS`, respectively. This shows the actual number of apps that had the capability to access this data and also confirms the decrease in the number of apps that could access children's contact information when compared to the results of Reyes's et al. [116] investigation in 2018. While only 8% of the apps we tested declared the `READ_PHONE_STATE` permission, 30% of the apps tested by Reyes et al. [116] did so. Similarly, Reyes et al. [116] found the `GET_ACCOUNTS` permission declared in 13% of the apps they tested whereas our results show that only 1.6% of those in our corpus did so (see Table 4 in Appendix B).

Thus, the takeaway from this analysis is that while we found evidence of the decrease in the number of apps that transmitted or were able to access personal data, third parties were still able to track children (RQ1). This was likely caused by the use of third-party SDKs that triggered the transmission of device identifiers.

5.2 TLS Usage

We also investigated the extent to which the apps secure their inbound or outbound communications using the Transport Layer Security (TLS) protocol. Only 51 of the 7,377 apps contained personal data in their inbound or outbound communications that were not secured using TLS (SSAIDs: 18 apps; location data: 16 apps; AAIDs: 12 apps; FIDs: 10 apps; and WiFi MAC: 1 app). For 22 of them, personal data was transmitted to first-party recipients only, whereas the recipients of personal data for the 29 remaining apps were third parties such as InMobi [78] and Umeng [128] (two of these apps contained personal data in their unencrypted communications to first-party servers as well). For location data, only one app included fine GPS coordinates, whereas the remaining 15 apps only contained IP location in their unencrypted communications. Twenty-one of the 51 apps displayed DFF badges, whereas 6 were teacher-approved. While these percentages show that Google Play was still publishing apps that were not fully utilizing TLS, they also suggest that the majority of apps were doing so. Thus, finding that less than 1% of apps did not fully adopt TLS shows an increase in the adoption of TLS in the past few years (RQ1), as Reyes et al. [116] showed that 40% of child-directed apps suffered from communication insecurity due to not using TLS.

5.3 Effects of Platform Policies

In this section, we analyze how Google Play's recent policies impacted developers' use of third-party SDKs in child-directed apps and app set IDs as an alternative to other device identifiers. We also evaluate the extent to which DFF-badged apps were compliant.

5.3.1 Use of Third-party SDKs. We observed the use of several third-party advertising SDKs that were not on Google's list of self-certified SDKs [77], including SDKs provided by Meta [99], Yandex [135] and Start.io [124]. Other apps shared personal data with SDK providers who do not allow using their SDKs in child-directed apps (e.g., Flurry [62, 63]).

In the following, we analyze developers' adoption of third-party SDK privacy configurations in child-directed apps (RQ2). These include server- or client-side configurations aimed at communicating to third-party SDK servers whether: (1) children are among apps' target audiences, (2) COPPA, GDPR or CCPA provisions are applicable, (3) users provided consent to data sharing or sale of personal information, or (4) developers instructed the SDK to stop

Table 2: Number of unique apps that transmitted personal data across all the tests.

	Data type	# of apps [all app versions] (N=7,377)	# of apps [most recent app versions only] (N=7,377)	Reyes et al. [116] (N=5,855)
Apps	Android Advertising IDs (AAIDs)	824 apps (11.2%)	652 apps (8.8%)	3454 apps (59%)
	Android IDs (SSAIDs)	720 apps (9.8%)	611 apps (8.3%)	NA
	Geolocation data (excluding IP location)	7 apps (0.09%)	7 apps (0.09%)	184 apps (3%)
	WiFi MAC	2 apps (0.03%)	1 app (0.01%)	NA
	Router BSSIDs	2 apps (0.03%)	2 apps (0.03%)	101 apps (2%)
	Router SSIDs	2 apps (0.03%)	2 apps (0.03%)	148 apps (2.5%)
	Names	1 app (0.01%)	1 app (0.01%)	NA
	E-mail addresses	1 app (0.01%)	1 app (0.01%)	107 apps (1.8%)
	Phone numbers	none	none	10 apps (0.17%)
	App set IDs (N=4,975)	562 apps (11.3%)	562 apps (11.3%)	NA
	Firebase installation ID (FIDs)	2115 apps (29%)	2080 (28.2%)	NA
GMS/GSF	AAIDs	702 apps (9.5%)	666 apps (9%)	NA
	SSAIDs	8 apps (0.11%)	7 apps (0.09%)	NA
	FIDs	579 apps (8%)	556 apps (7.5%)	NA

collecting certain data types (e.g., AAIDs). Detailed measurements are provided in Tables 5 and 6 in Appendix D.

Configurations for Child-Directed Treatment. We examined the use of COPPA-related configurations in apps that shared personal data with third-party servers contacted by the SDKs we tested (Section 4.3). Given that Google's policies prohibit collecting AAIDs from children [21], we focused on understanding whether developers correctly used privacy configurations for third-party SDKs embedded in their apps and whether correct usage of such configurations affected sharing of AAIDs with third parties (Section 4.3). We found Meta's SDK [99] within 24 apps, 20 of which transmitted AAIDs while incorrectly setting Meta's `COPPA` flag [95] to `false` and none set it to `true`. However, we observed higher levels of adoption of COPPA-related configurations offered by a number of other advertising SDKs, including ironSource [85], Unity [54], Google AdMob [3], and AdColony [2]. Of the 299 apps that communicated with ironSource [85], 94% correctly set `is_coppa` or `is_child_directed` to `true`, which are based on mandatory server-side and optional client-side configurations, respectively [82]. Of the 56 apps that transmitted AAIDs to ironSource [85], 73% used at least one of these configurations correctly, 16% used them incorrectly, and 11% did not use them. Developers who use ironSource [81, 85] as an ad mediation platform are also offered a set of optional privacy signals that can be propagated to their selected ad provider (e.g., `Applovin_AgeRestrictedUser` for developers who instruct ironSource [81, 85] to request ads from AppLovin [11]). We identified 191 apps that correctly used at least one of these configurations and none incorrectly did so (see Table 5).

Of the 741 apps sending data to Unity [54], 61% signaled that they were child-directed through a server-side configuration. Only 20 of these correctly-configured apps transmitted AAIDs to Unity, whereas the remaining apps that did so either signaled that they were not child-directed (36) or did not set a signal in their transmissions (73). Sixty-five percent of apps that communicated with InMobi [30, 78] similarly signaled that COPPA applies to them, 15% of which transmitted AAIDs to InMobi [78]. For Google AdMob [3], we monitored the use of `setTagForChildDirectedTreatment` and the related `setTagForUnderAgeOfConsent`, which are client-side configurations. The values of `tag_for_child_directed_treatment` and `tag_for_under_age_of_consent` are set based on these configurations, respectively. We found that at least one of these configurations was used correctly by 37.5% of the 3,147 apps that communicated with `doubleclick.net` or another of Google's domains, which is `fundingchoicemessages.google.com`.[8] Of the 64 apps that shared the AAID with one of these endpoints, 32 apps set one or both of these signals to their correct values, 29 apps signaled that they were not child directed, and 3 apps did not use any of them.

For AdColony [2], we monitored the use of `coppa_required` and `is_child_directed`, which are set client side, and a `coppa` signal based on a server-side configuration [2]. Of the 158 apps that communicated with AdColony, 93% correctly used at least one of these configurations. Of the 34 apps that transmitted AAIDs to AdColony, 26 set `coppa_required` or `coppa` to `true`, 8 incorrectly set one or both to `false`, and none set `is_child_directed` to `true`, which prevents transmission of AAIDs [1]. The same configuration was correctly used by another 109 apps to disable collecting AAIDs from children. Similarly, we searched the data sent to Vungle [131] by 203 apps to measure adoption of its `is_coppa` signal, which is based on an optional client-side configuration [130], and found it to be correctly set to `true` by 82%. None of the apps that correctly used this configuration transmitted AAIDs to Vungle, whereas 5 misconfigured apps did so. Chartboost's `addDataUseConsent` function [31, 51] to set `coppa` to `true` in outbound transmissions had a similar effect, as none of the 39 apps that did so transmitted AAIDs to Chartboost [31]. Additionally, none of the apps that communicated with Yandex [135] or Appodeal [12] used their privacy configurations correctly; neither of which are on Google's list of

[8] In certain cases, a signal was sent to one of these endpoints (e.g., `doubleclick.net`), but AAIDs were sent to the other (e.g., `fundingchoicemessages.google.com`). Since both of them belong to Google and AdMob's User Messaging Platform SDK [5] communicates with the latter, we assumed that they are related.

certified SDKs [77]. Furthermore, although AppLovin [11] is no longer on this list [77] and that integrating it in apps that are primarily child-directed is against its terms of service [10], 49 apps shared AAIDs with AppLovin, only 6 of which signaled that they are child-directed through a client-side configuration.

After comparing the number of apps with COPPA-related signals set to correct values in their transmissions with those that signaled that they were not child-directed, we identified 72 apps that set different signals to conflicting values. They either set COPPA-related signals that belong to the same SDK to conflicting values or did so for signals that belong to different SDKs embedded in their apps. We additionally found developers who correctly used COPPA-related signals in some of their apps, and at the same time incorrectly used them in other of their child-directed apps. These cases demonstrate the difficultly developers are experiencing with configuring SDKs for child-appropriate treatment.

In 2018, Reyes et al. [116] quantified the use of COPPA-related configurations provided by Meta [99] and Unity [54]. While they found correct usage of Unity's COPPA configuration [54] in only 16.5% of the apps that used Unity, we show that this percentage increased to 61% (RQ1). Although our corpus included only 24 apps that embedded Meta's advertising SDK [99], none of these apps set its `COPPA` flag to `true`. However, Meta prohibits use of its SDK in apps that are primarily child directed; the `COPPA` flag is provided to identify child users of mixed-audience apps [95]. Therefore, this suggests an improvement in apps' privacy behaviors (i.e., a decrease in the number of apps that used Meta's advertising SDK).

Configurations for CCPA Compliance. We observed varying levels of adoption of configurations that instruct third parties not to sell users' personal data to other parties. Of the apps that communicated with AdColony [1], Vungle [130], and ironSource [82], 72%, 73% and 68.5% opted out children residing in California from data sales, respectively. Related configurations offered by Unity [52, 54], Chartboost [31, 51], InMobi [30], Google AdMob [3, 4], AppLovin [10], and Flurry [63] received lower adoption (see Table 6). However, some of these SDKs offered server-side configurations that did not result in transmitting discernible signals in inbound traffic, and thus our results can be considered a lower bound for the adoption of these configurations. Furthermore, it is unclear whether apps that failed to correctly use these configurations were subject to compliance with the CCPA [102]. Answering this question might require estimating the sizes of developers' organizations, their revenues or the number of California residents who used their apps, which we leave to future work. We also found signals that communicated that users consented to data sharing when they did not. Since most of them were offered for GDPR compliance purposes and we tested apps from California, we leave verifying this observation to future work aimed at testing child-directed apps from the EU.

Other Configurations. We investigated the adoption of configurations that result in requesting non-personalized ads or preventing the transmission of personal data. Unity [54] offers a server-side option that allows disabling personalized advertising when indicating that an app is not child-directed through a server-side control. The `user.nonBehavioral` is another signal, which serves a similar purpose but can be set using a client-side configuration [54]. None of the apps that incorrectly used Unity's COPPA configuration disabled personalized advertising using any of these controls. Meta's analytics SDK [96] offers a client-side control that prevents transmitting AAIDs, and which results in setting a signal called `advertiser_id_collection_enabled` in exchanged communications. This was correctly used by only 22% of the apps that embedded this SDK. Similarly, 75% and 87% of apps that communicated with ironSource [82] and Vungle [130], respectively, disabled collecting AAIDs. We additionally found Google AdMob's `npa` flag correctly used by 342 apps to disable personalized advertising, 71% of which correctly used a COPPA-related signal (e.g., `tfcd`) and only 6% are of those that signaled that they are not child-directed.

The general takeaway from our analysis of third-party SDK configurations is that developers were clearly not using all the privacy configurations that were offered to them by third-party SDKs, and also did not consistently use them in all of their apps.

SDK Versions. Google also specifies the minimum versions of third-party advertising SDKs that can be embedded in children's apps [77]. For six of the SDKs that were included on Google's list [77], we investigated whether allowed versions were used. We found that 31%, 28% and 10% of the apps that integrated Chartboost's [31], Unity Ads's [54], and InMobi's [78] SDKs, respectively, used a version below the minimum version allowed [77]. The same applies to 8%, 5% and 4% of apps that integrated AdColony's [2], Vungle's [131], and ironSource's [85] SDKs, respectively.

5.3.2 App Set IDs. The app set ID is an identifier that Google recently introduced to allow obtaining analytics about app usage while preventing cross-device tracking or long-term tracking across apps released by different developers [16, 40, 47, 48]. Unlike AAIDs, which are the same for all apps installed on the same device until they get reset by users using system settings, app set IDs can only uniquely identify transmissions from apps released by the same developer on Google Play that are installed on the same device [40, 47, 48]. Unlike SSAIDs, which can similarly identify apps associated with a specific developer on the same device, the values of app set IDs change to new values once users uninstall all apps that share a common app set ID value, whereas changing SSAIDs requires users to factory reset their devices [43, 48]. Google's policies prohibit using app set IDs to target ads to child users and collecting it alongside certain other types of personal data [20, 73].

Of the 4,975 apps that we tested in the second round, 562 transmitted app set IDs. However, 30% of these 562 apps generated outbound transmissions that shared app set IDs and AAIDs (107) (see Figures 4 and 5), SSAIDs (67) or FIDs (1) with the same data recipients. The top data recipients in these cases were third-party SDK providers, such as SupersonicAds [85] (81), AdColony [2] (30), InMobi [78] (26), AppLovin [11] (23), and Chartboost [31] (16). While 69% of the 562 apps did not include additional identifiers in transmissions that contained app set IDs, 12% of these 389 apps did not consistently use app set IDs as replacement to other identifiers in all their transmissions, since they included AAIDs or SSAIDs in other transmissions. These findings show that while apps were starting to use app set IDs as an alternative to other device IDs, not all apps were utilizing its privacy properties (i.e., many apps gave third parties the ability to link app set IDs with other data types).

5.3.3 Designed for Families Badge. While complying with Google's families policies is mandatory for all apps that target children, developers may optionally include the "Committed to follow the

Play Families Policy" badge within their privacy labels [21, 22]. We examined whether having the badge displayed implied apps' compliance with Google Play's policies. Of the 3,684 apps that used the badge, we found:

- 698 apps (19%) transmitted AAIDs, 361 (10%) of which did so to domains other than `app-measurement.com`;
- 449 apps (12.2%) targeted Android 13 and declared the `AD_ID` permission;
- 340 apps (9.2%) transmitted AAIDs alongside app set IDs, FIDs or SSAIDs to the same recipients, 106 (2.9%) of which did so to domains other than `app-measurement.com`;
- 163 apps (4.4%) transmitted AAIDs to third-party advertising or analytics services without using any of their COPPA or CCPA privacy configurations correctly;[9]
- 145 apps (4%) prepared data safety labels that included at least one disclosure mistake about the personal data they transmitted;
- 97 apps (2.6%) declared the `ACCESS_FINE_LOCATION` permission;
- 89 apps (2.4%) transmitted AAIDs, SSAIDs, or app set IDs to providers of third-party advertising SDKs, which are not on Google's list of self-certified advertising SDKs [77] (AppLovin [11] and Yandex [135]);
- 71 apps (1.9%) embedded a version of a self-certified advertising SDK (Unity Ads [54], AdColony [2], ironSource [85], Chartboost [31], InMobi [78] or Vungle [131]) that Google Play prohibits embedding in child-directed apps [77]; and
- 21 apps (0.6%) did not use TLS in all their communications that contained personal data.

These percentages show that DFF-badged apps were not necessarily compliant with Google Play's families policies [21].

5.4 Data Safety Labels

We analyzed the data safety labels of the 828 apps that transmitted AAIDs, fine or coarse geolocation coordinates, WiFi MAC addresses, router SSIDs, router BSSIDs, names, or email addresses (see Table 2). Of these apps, 491 had disclosure issues in their labels. Of the 491 apps, 393 either did not have a label (284) (Figure 9) or had a label that explicitly stated that no personal data is collected or shared by their apps (109) (Figure 8). Of the 207 apps that posted inaccurate labels, 200 did not disclose at least one of the data types they transmitted and 121 apps did not have all the correct purposes for the data types they collected and/or shared listed in their labels. Additionally, 70% of the 207 apps had the DFF badge in their inaccurate labels. After excluding the 172 apps that we did not observe continuing to transmit certain data types in their later versions, the number of apps that had disclosure mistakes decreased from 491 to 399 (61% of 657). Table 3 summarizes how this improved over time for the apps that we observed transmitting AAIDs, location data, MAC addresses, router SSIDs or BSSIDs, names, and email addresses.

We investigated whether developers' use of "data is encrypted in transit" versus "data isn't encrypted" in their labels was accurate.

[9]We assumed that a client- or server-side privacy configuration was not used when we did not find the corresponding privacy signal in inbound or outbound transmissions.

Table 3: Disclosure inaccuracies in privacy labels.

	# of apps (N=828) [all app versions]	# of apps (N=657) [recent versions only]
Total number of apps that have DSL mistakes	491	399
# of apps that did not have DSLs	284	259
# of apps that did not disclose a data type	200	134
# of apps that did not disclose a purpose	121	96

Of the 51 apps that did not use TLS in transmissions that contained personal data (Section 5.2), only 20 apps disclosed that encryption was not used, whereas the rest either incorrectly stated that encryption was used when it was not (9), or did not make any disclosures about whether data was encrypted in transit (22). Five of the 20 apps initially disclosed that encryption was used when it was not, but updated their labels to state that it was not used instead of securing their unencrypted communications.

We examined whether the 2,578 apps that transmitted SSAIDs, app set IDs, and FIDs to domains other than `app-measurement.com` disclosed doing so in their labels and found that 57% (1,472) of them did not, 53 of them updated their labels to disclose transmitting "device IDs," whereas 42 removed this data type from their labels after our initial tests. Of the 2,831 apps that transmitted any of the data types in Table 2 to domains other than `app-measurement.com`, 59% either did not have label or had one that did not disclose all the data types they transmitted (RQ2). Fifty-seven of these apps fixed their labels after the first testing round to add the relevant disclosures and 41 removed mention of device identifiers, despite continuing to transmit them during our testing.

The improvement in apps' privacy practices over time was also corroborated by observing that the number of apps without labels decreased from 2,189 in the first testing round to 499 in the second round. Of the apps tested in both rounds, 177 of the apps that initially did not have a label updated their app listing to publish one. Thus, the total number of apps that did not have a data safety label in any of the two testing rounds was 2,071. The improvements observed between the two testing rounds could have resulted from developers reacting to Google Play's notifications that informed them that their labels *"will be invalidated"* unless they include accurate disclosures by specific enforcement deadlines [122].

Our results also suggest that developers were not always able to correctly follow Google Play's classification of data types or purposes, or understand Google Play's distinction between *collected* versus *shared* data types [22] when preparing their labels. For example, we found developers who transmitted AAIDs to advertising services while not considering doing so as data sharing, or used "app functionality" as a purpose for doing so instead of using "advertising or marketing." Additionally, 113 of the apps that transmitted device identifiers used "user IDs" instead of "device IDs" in their labels, which could suggest that their developers were unable to distinguish between these two data types.

6 Developer survey

We recruited developers of child-directed apps to participate in a follow-up survey that focused on understanding developers' experiences with configuring third-party SDKs for privacy compliance and preparing their apps' privacy labels. It also asked about the privacy guidance that they rely on (if any) or need to satisfy their compliance obligations (see Appendix A). It also helped us shed light on the phases of the development process where privacy guidance is most likely to be used and the specific privacy guidance formats that developers would like to have (e.g., privacy checklists, app templates, and interactions with experts).

We used emails found in apps' Play Store listings to invite 2,378 developers to participate in the survey and sent invitations to developers through their publicly-available accounts on social media. The contacted developers were part of app development organizations whose child-directed apps were available on Google Play. To incentivize participation, we offered respondents the option to enter a drawing for one of five $200 Amazon gift cards. We received 53 complete responses to our survey. This response rate is consistent with the rates observed in related studies that sent surveys to the same population (e.g., [6, 56]).

6.1 Results

This section summarizes the results of our survey. For open-ended responses, two researchers qualitatively coded them using a codebook that they jointly developed after analyzing a subset. During this process, they frequently met to discuss coding results and resolve disagreements until they reached a Cohen's κ score of 0.90.

Working with Third-Party SDKs. Most of the respondents indicated that their apps embedded SDKs that shared personal data with some of the third parties listed in Tables 5 and 6 for advertising or analytics purposes. Of the 53 respondents, 49%, 20%, 15%, and 6% used SDKs in their apps that shared data with Google AdMob [3], Unity Ads [54], Meta [96, 97], and ironSource [85], respectively. While 74% of respondents indicated familiarity with third-party SDK privacy configurations, only 55% used them in some or all of their apps (RQ3). Those who did not use privacy configurations despite their familiarity with them provided a number of justifications. They indicated that either Google Play accepted their apps regardless, there is no need to configure SDKs included on Google Play's list of certified SDKs [77] for compliance, the SDKs were compliant by default, or that they did not understand how to do so. Furthermore, 64% and 69% of respondents found configuring SDKs for compliance or keeping them up-to-date to be challenging, respectively. They provided a number of reasons that can explain why not all of the apps we tested used privacy configurations correctly (Tables 5 and 6). These include shortcomings in SDK developer documentation that led developers to be unable to identify or correctly use the specific configurations applicable to their apps and the operational overhead associated with keeping up with the frequent updates made by SDK providers to these configurations. One respondent explained: *"you have to read all the manuals and find some information in the bottom spots of some instructions which make creating these environments troublesome"* (R46).

We also examined whether they understood the consequences of not using privacy configurations and found that 40% said that their apps could get removed from the store as a result. This finding further demonstrates the power of app platforms in shaping developers' perceptions about the consequences of not addressing privacy issues, which might subsequently lead them to improve their privacy practices. However, 45% trusted that the data collection practices of third-party SDKs would not introduce privacy compliance issues. This could therefore explain why privacy issues that result from developers' use of third-party SDKs might not get addressed until developers get notified by the store about them.

Preparing Privacy Labels. Thirty-percent of respondents found the task of preparing data safety labels to be difficult. While 57% of respondents' organizations dedicated parts of their development processes to this task (RQ3), 60% did not employ traffic analysis tools to identify data types transmitted by their apps before submitting their labels. Furthermore, 45% tasked development teams with filling out Google Play's data safety questionnaire, whereas the rest assigned this task to non-technical teams. This could explain the observed discrepancies between the actual behaviors of the tested apps and their labels (Section 5.4). Two main approaches were relied on for guidance on what to include in data safety labels, which were third-party SDK documentation and notifications received from Google Play. One explained: *"Sometimes we know the answer, sometimes we choose at random...and then we wait for a decision from Google Play. If something is wrong, fix it and re-upload"* (R31).

Only 21% of respondents stated that Google Play does not check the accuracy of disclosures included in privacy labels. Fifty-three percent of respondents, however, believed that these labels should be automatically prepared by the store. Respondents expressed challenges with mapping their apps' data practices to Google Play's classification of data types [22], frequently maintaining their labels to reflect changes in their apps' practices, and identifying the practices of third-party SDKs that need to be reflected in their labels. This is supported by the analysis presented in Section 5.4, as most of the observed discrepancies were caused by sharing of personal data with third parties. One participant explained: *"Even Google SDKs like AdMob don't provide specific answers about how to fill out the Google data safety sheets. It is difficult to find out how to do it correctly"* (R18).

Access to Privacy Guidance. Only 42% of respondents indicated that their organizations provided them with guidance on how to fulfill their privacy compliance obligations. This was mostly presented in checklists, templates for privacy policies, or descriptions of compliance requirements. Additionally, 81% expressed their willingness to add better support for privacy compliance in their processes once they are provided with proper guidance that can help them do so. This included having privacy checklists, reference apps demonstrating how to avoid common privacy issues, and the ability to interact with experts who can provide specific technical or legal advice. Many also needed guidance on how to avoid compliance issues that result from embedding third-party SDKs. Specifically, they needed to be guided through how to use SDK privacy configuration options, how to understand whether collected data would be used for prohibited purposes, and how to identify the types of personal data collected by SDKs. This provides further support for the results of our technical analyses, which showed that most of the detected issues were due to improper use of third-party SDKs. For example, one respondent explained that they wanted a *"list of*

identifiers used by the SDK and reference to relevant regulations for their use" (R38).

As for when in their development processes they would be most likely to use this guidance, most respondents indicated that they would use it during the development (58%) or testing (36%) phases of their processes. However, 57% also preferred to have Google Play present this guidance as part of the app submission process. This provides further evidence of the powerful impact that platforms could have on developers' privacy compliance processes.

The results of this survey therefore show that many developers added support for configuring third-party SDKs for compliance in their development processes and were adapting to Google Play's requirement that asked them to prepare data safety labels. However, they were also facing challenges with these tasks and needed to be guided through how to correctly use these privacy controls.

7 Discussion

This study showed a drastic decrease in the number of apps that transmitted children's personal data or did not encrypt their communications. However, it also showed that use of third-party SDKs was still causing many types of privacy issues.

Third-Party SDK Developers. Misuse of third-party SDKs led to developers not accurately disclosing their apps' practices and the sharing of children's personal data. While Google Play made major updates to its policies in the past few years [112], these policy efforts are still not eliminating privacy issues caused by the use of third-party SDKs. This is also limiting the efficacy of Google Play's recently-introduced data safety labels [22] that aim to provide users with higher levels of transparency. Developers' lack of understanding of the behaviors of third-party SDKs embedded in their apps not only allowed sharing of personal data, but also enabled linking of identifiers, over-privileging apps, and publishing inaccurate data safety labels (Sections 5.1 and 5.4).

Currently, third-party SDKs are not aligning their practices with Google Play's policies [109]. While the policies prohibit collecting AAIDs from children, correct use of their privacy configurations did not necessarily prevent collecting AAIDs from children or linking them with other identifiers (see Figures 4 and 5, and Tables 5 and 6). Use of third-party SDKs also led to sharing app set IDs with advertising services even though the policies do not allow using them for advertising purposes [20, 73]. Such shortcomings in the designs of third-party SDKs can lead to privacy compliance issues, as we demonstrate, even if SDKs are correctly configured for compliance.

Improvements can (and should) be made to the design of third-party SDK privacy configurations to reduce the operational overhead associated with using them. These include reducing the number of configuration options that developers need to use for each SDK and also standardizing them across the industry. As depicted in Tables 5 and 6, third-party SDKs currently have different configuration options that can be used for different purposes, which increases the likelihood of misconfiguring them. Before using ironSource's SDK [85], for example, developers must decide on which of its two server-side and five client-side options to use to comply with COPPA, GDPR, and the CCPA [82]. To reduce the number of configurations, third-party SDK providers could utilize publicly-available metadata about apps (e.g., whether they are DFF-badged or teacher-approved) to determine whether they are primarily child-directed and disable targeted advertising accordingly (i.e., without developers needing to take action). This feature is currently partly supported by Google AdMob [3], but not by other SDK providers. Many privacy issues could also be eliminated once third-party SDKs use privacy-preserving defaults. While we showed an overall improvement in apps' privacy practices, more progress can therefore be achieved once third-party SDK providers do their part in helping reduce the burden of compliance on developers.

The Google Play Store. The observed overall improvement in apps' privacy practices is likely the result of developers adapting to changes made to Google Play's policies over the past few years [109, 112]. This shows that Google Play is in a powerful position to influence organizational privacy practices worldwide. Given the various privacy issues that are still present in child-directed apps, there is a need for further strengthening Google Play's enforcement efforts to protect children's privacy. Google Play might also need to utilize its powerful position to educate developers about how privacy compliance issues can be identified and addressed early in their development processes. To do so, the various points of interaction between Google Play and developers could be leveraged to raise developers' awareness about how to build privacy-preserving apps. These are mainly Google Play's developer console [108], e-mail notifications, and policy documentation [109].

Despite the drastic reduction in the number of child-directed apps that transmitted personal data, improper use of third-party SDKs is still the leading root cause of privacy issues. While Google Play is continuously improving its developer policies, it appears to not be comprehensively auditing the practices of third-party SDKs before including them on its list of certified SDKs [25, 77]. Instead of requiring SDK developers to *self-certify* their compliance with Google Play's policies [25], the certification process should involve performing technical investigations of third-party SDKs' data collection practices from multiple perspectives. There are many heuristics that can be employed in this process, which include examining the effects of using SDK privacy configurations on data collection, understanding the extent to which SDKs are able to link different data types, and detecting whether SDKs are using side channels to access personal data [115]. Employing a comprehensive auditing process of SDKs' privacy practices is therefore likely to improve the effectiveness of Google Play's enforcement efforts.

Anecdotal evidence suggests that Google Play has been auditing apps' data transmission practices to verify the accuracy of data safety labels [121, 122]. As part of its ongoing enforcement efforts, developers have been receiving notifications requiring them to update their labels once discrepancies are detected. While this approach is likely leading many developers to fix disclosure issues detected in their labels, following a proactive approach to perform such audits might be more effective. Google Play can incorporate features that provide developers with real-time feedback about their apps' privacy practices as part of the app submission process [108]. This would prevent developers from publishing apps that have privacy issues, allow them to understand their apps' data practices, and resolve privacy issues early in their development processes. We showed that 60% of survey respondents did not have access to traffic analysis tools (Section 6.1), which emphasizes the need for providing developers with this type of feedback.

Google Play's developer console [108] could also utilize data entered during the app submission process to provide customized privacy guidance. While Google Play has access to data about apps' target audiences and the countries where apps are available to users, this data could be utilized to raise developers' awareness about their compliance obligations. Google Play's enforcement efforts should also prevent developers from making inaccurate representations to users through its badging system. For example, while the DFF badge allowed developers to communicate their *commitment* to complying with Google's policies [21, 22], we showed that even DFF-badged apps suffered from privacy issues (Section 5.3.3).

Regulators and Policy Makers. Most of the data sharing issues resulted from failures to apply data minimization principles by third-party SDKs. Sharing of personal data even when privacy configurations were used correctly and including different identifiers in the same transmissions are two prominent examples. This calls for the need of adding more restrictions to existing legal provisions that allow collecting device IDs for certain purposes. For instance, restricting the number of identifiers that can be collected by third parties who are relying on COPPA's internal operations exception could lead to further improvements in overall privacy practices [34].

Our results showed that Google Play's policy efforts can shape privacy practices of development organizations publishing apps from various countries of the world. Measuring adoption of different types of SDK privacy configurations also demonstrated how developers are responding to requirements of federal (i.e., COPPA) versus state regulations (i.e., CCPA) in the US. However, the effects of Google Play's efforts on worldwide developer privacy practices raise the question of whether enacting more privacy regulations in various regions of the world is introducing unnecessary complexity to development processes. Given the differences in the regulatory requirements of privacy regulations, unifying privacy requirements of different privacy regulations enacted in one country (e.g., federal and state regulations in the US) or across countries is likely to reduce the burden of compliance on developers.

While our investigations applied specifically to child-directed apps, we still show that app platforms are in a powerful position to make drastic improvements in developers' privacy practices. Future work could therefore explore the feasibility of aligning platform policies with the requirements of applicable privacy regulations and the specific roles that app platforms can play in enforcing the requirements of privacy regulations on all app populations.

8 Limitations

We used technical and qualitative methods to understand how platform policy changes impact development practices. However, a few factors could have introduced a degree of uncertainty to our results. First, while we also considered apps that are not DFF-badged, all developers of child-directed apps are required to comply with the families policies [21], just as they are required to comply with COPPA. For this reason, and because we wanted to identify as many child-directed apps as possible, comparing our results to those of Reyes et al. [116] is slightly confounded because Reyes et al. only examined apps that were enrolled in DFF (i.e., a subset of apps subject to COPPA). Second, traffic obfuscation could have affected our analysis of traffic data. However, we employed advanced de-obfuscation techniques that were verified in the literature, and therefore our results can be considered a lower bound for the types of privacy issues that exist in the general population of child-directed apps. Third, there could have been privacy configurations that we did not consider due to: (1) developers using old SDK versions that offered different configurations or (2) SDKs employing server-side configurations that do not trigger sending privacy flags. To address this limitation, we repeatedly monitored third-party SDK documentation to test the updates made to their privacy configurations over the prior two years and tried to identify other indicators of the use of server-side configurations (e.g., not transmitting AAIDs when used correctly). Fourth, although Google's list of self-certified SDKs [77] was updated after our pilot tests to specify the minimum allowed SDK versions and removed AppLovin [11], we included these in our measurements to understand developers' privacy practices at that point in time.

9 Ethics

The survey we distributed to app developers was IRB-approved by our university. To preserve respondents' privacy, we did not collect identifiable data from them other than an email address that we stored separately from the survey data. We also allowed developers to participate without providing their email addresses. Participation in our survey was voluntary and informed consent was obtained from all the respondents. For recruitment, we used the publicly available email addresses posted by developers on their app listing on the Google Play Store. When we sent our email invitations, we included an option that allowed developers to opt out from any future communications from us. While our large-scale crawling of the Google Play Store to identify and test child-directed apps could have put some pressure on the store, our crawling approach is consistent with those followed in prior related studies (e.g., [116]). Furthermore, since apps were downloaded over a long period of time (i.e., several weeks) using several Android devices, it is unlikely that our approach negatively impacted the store.

10 Conclusion

We investigated the privacy practices of child-directed apps that were available on Google Play in 2023. Based on comparisons of our results with those observed in 2018, we found evidence of continuous improvements in the privacy behaviors of these apps. This is likely due to the changes made to Google Play's policies over the past few years and developers becoming increasingly aware of their compliance responsibilities. However, we also show that the community is yet to address one of the leading causes of privacy issues, which is developers' inabilities to use third-party SDKs in ways that do not negatively affect their compliance with platform policies and applicable privacy laws. This is contributing to the persistence of the same types of privacy issues that were observed in 2018. It is also negatively affecting the effectiveness of app set IDs and leading to the prevalence of disclosure issues in privacy labels, both of which are recently-introduced interventions aimed at providing users with higher levels of privacy. Furthermore, while adoption of third-party SDK privacy configurations is increasing, developers are still struggling to correctly use them across their apps. The results of a follow-up developer survey showed that while developers

were making changes to their development processes in response to recently-introduced platform policies, use of third-party SDKs was still leading to the presence of privacy issues. Taken together, our findings demonstrate the need for providing developers with actionable privacy guidance and shed light on how Google Play can further strengthen its policy enforcement efforts.

Acknowledgments

This work was supported by the U.S. National Science Foundation under grants CNS-2247951 and CCF-2217771; the AEI PARASITE project, funded by the Agencia Estatal de Investigación (MICIU/AEI/10.13039/501100011033) under Grant Agreement PID2022-143304OB-I00; and the Natural Sciences and Engineering Research Council of Canada (NSERC; funding reference number RGPIN/04237-2018). Dr. N. Vallina-Rodriguez's work has been partially supported by the 2019 Ramon y Cajal fellowship program, funded by the MICIU/AEI /10.13039/501100011033 and the FSE Invierte en tu futuro, under grant agreement RYC2020-030316-I. We would additionally like to thank Chris Hoofnagle for the invaluable feedback he provided on this paper and Chris Harjadi for helping with our qualitative coding of the survey data. This work was made possible by Refjohürs Lykkewe.

References

[1] AdColony. 2023. AdColony's Privacy Settings. Retrieved January 3, 2024 from https://github.com/AdColony/AdColony-Android-SDK/wiki/Privacy-Laws
[2] AdColony. 2023. AdColony's SDK (deprecated). Retrieved November 17, 2023 from https://support.adcolony.com/
[3] Google Admob. 2024. Targeting Configurations. Retrieved November 11, 2024 from https://developers.google.com/admob/android/targeting
[4] Google Admob. 2024. U.S. states privacy laws compliance. Retrieved November 11, 2024 from https://developers.google.com/admob/android/privacy/us-states
[5] Google Admob. 2024. User Messaging Platform SDK. Retrieved November 11, 2024 from https://developers.google.com/admob/android/privacy
[6] Noura Alomar and Serge Egelman. 2022. Developers say the darnedest things: Privacy compliance processes followed by developers of child-directed apps. *Proceedings on Privacy Enhancing Technologies* 4, 2022 (2022), 24.
[7] Amplitude. 2024. Amplitude Services. Retrieved November 11, 2024 from https://amplitude.com/
[8] Android. 2024. Google Mobile Services (GMS). Retrieved November 11, 2024 from https://www.android.com/gms/
[9] APKPure. 2024. Retrieved November 16, 2024 from https://apkpure.com/
[10] AppLovin. 2023. Consent, Other Applicable Flags, and Data APIs. Retrieved November 11, 2024 from https://dash.applovin.com/documentation/mediation/android/getting-started/privacy
[11] AppLovin. 2024. *Support Center.* Retrieved November 11, 2024 from https://dash.applovin.com/documentation/mediation
[12] Appodeal. 2024. Appodeal Android SDK. Retrieved November 11, 2024 from https://docs.appodeal.com/android/get-started
[13] AppsFlyer. 2024. AppsFlyer Android SDK. Retrieved November 11, 2024 from https://dev.appsflyer.com/hc/docs/android-sdk
[14] AppStore. 2024. *App Review Guidelines.* Retrieved November 11, 2024 from https://developer.apple.com/app-store/review/guidelines/
[15] Reuben Binns, Ulrik Lyngs, Max Van Kleek, Jun Zhao, Timothy Libert, and Nigel Shadbolt. 2018. Third party tracking in the mobile ecosystem. In *Proceedings of the 10th ACM Conference on Web Science*. 23–31.
[16] Android Developers Blog. 2021. *Announcing Policy Updates To Bolster Privacy and Security.* Retrieved November 11, 2024 from https://android-developers.googleblog.com/2021/07/announcing-policy-updates-to-bolster.html
[17] Android Developers Blog. 2022. *Expanding Play's Target Level API Requirements to Strengthen User Security.* Retrieved November 11, 2024 from https://android-developers.googleblog.com/2022/04/expanding-plays-target-level-api-requirements-to-strengthen-user-security.html
[18] Duc Bui, Yuan Yao, Kang G Shin, Jong-Min Choi, and Junbum Shin. 2021. Consistency Analysis of Data-Usage Purposes in Mobile Apps. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 2824–2843.
[19] Google Play Console Center. 2023. *Ads.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/9857753
[20] Google Play Console Center. 2023. *User Data.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/10144311
[21] Google Play Policy Center. 2023. *Google Play Families Policies.* Retrieved July 10, 2023 from https://support.google.com/googleplay/android-developer/answer/9893335
[22] Google Play Policy Center. 2023. *Provide information for Google Play's Data safety section.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/10787469
[23] Google Play Policy Center. 2023. *User Data.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/13316080
[24] Google Play Policy Center. 2024. *Deceptive Behavior.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/9888077
[25] Google Play Policy Center. 2024. *Families Self-Certified Ads SDK Program.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/9900633
[26] Google Play Policy Center. 2024. *Policy Deadlines.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/table/12921780
[27] Google Play Policy Center. 2024. *SDK Requirements.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/13323374
[28] Google Play Policy Center. 2024. *Store Listing and Promotion.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/topic/9877064
[29] Google Play Store Policy Center. 2024. *Google Play's Target API Level Policy.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/11917020
[30] InMobi Support Center. 2024. InMobi - Android SDK Documentation. Retrieved November 11, 2024 from https://support.inmobi.com/monetize/sdk-documentation/android-guidelines/overview-android-guidelines
[31] Inc Chartboost. 2024. Chartboost Monetization and Advertising. Retrieved November 11, 2024 from https://www.chartboost.com/
[32] Federal Trade Commission. 2012. Mobile Apps for Kids: Current Privacy Disclosures are Disappointing. Retrieved January 21, 2024 from https://www.ftc.gov/sites/default/files/documents/reports/mobile-apps-kids-current-privacy-disclosures-are-disappointing/120216mobile_apps_kids.pdf
[33] Federal Trade Commission. 2012. Mobile Apps for Kids: Disclosures Still Not Making the Grade. Retrieved January 21, 2024 from https://www.ftc.gov/sites/default/files/documents/reports/mobile-apps-kids-disclosures-still-not-making-grade/121210mobilekidsappreport.pdf
[34] Federal Trade Commission. 2013. Children's Online Privacy Protection Rule (COPPA). Retrieved November 11, 2024 from https://www.ftc.gov/legal-library/browse/rules/childrens-online-privacy-protection-rule-coppa
[35] Federal Trade Commission. 2019. FTC Seeks Comments on Children's Online Privacy Protection Act Rule. Retrieved January 20, 2024 from https://www.ftc.gov/news-events/news/press-releases/2019/07/ftc-seeks-comments-childrens-online-privacy-protection-act-rule
[36] Federal Trade Commission. 2020. Complying with COPPA: Frequently Asked Questions. Retrieved November 11, 2024 from https://www.ftc.gov/business-guidance/resources/complying-coppa-frequently-asked-questions
[37] Erik Derr, Sven Bugiel, Sascha Fahl, Yasemin Acar, and Michael Backes. 2017. Keep me Updated: An Empirical Study of Third-party Library Updatability on Android. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2187–2200.
[38] Android Developers. 2023. Android 8.0 Behavior Changes. Retrieved November 11, 2024 from https://developer.android.com/about/versions/oreo/android-8.0-changes
[39] Android Developers. 2023. Android Asset Packaging Tool (AAPT). Retrieved November 11, 2024 from https://developer.android.com/tools/aapt2
[40] Android Developers. 2023. *AppSetId.* Retrieved November 11, 2024 from https://developer.android.com/design-for-safety/privacy-sandbox/reference/adservices/appsetid/AppSetId
[41] Android Developers. 2023. UI/Application Exerciser Monkey. Retrieved November 11, 2024 from https://developer.android.com/studio/test/other-testing-tools/monkey
[42] Android Developers. 2024. Android Debug Bridge (ADB). Retrieved November 11, 2024 from https://developer.android.com/tools/adb
[43] Android Developers. 2024. Android ID. Retrieved November 11, 2024 from https://developer.android.com/reference/android/provider/Settings.Secure#ANDROID_ID
[44] Android Developers. 2024. *Android Releases.* Retrieved November 11, 2024 from https://developer.android.com/about/versions
[45] Android Developers. 2024. Android SDK: Command line tools. Retrieved November 11, 2024 from https://developer.android.com/tools
[46] Android Developers. 2024. *Behavior changes: Apps targeting Android 13 or higher.* Retrieved November 11, 2024 from https://developer.android.com/

about/versions/13/behavior-changes-13
[47] Android Developers. 2024. *Best practices for unique identifiers.* Retrieved November 11, 2024 from https://developer.android.com/training/articles/user-data-ids
[48] Android Developers. 2024. *Identify developer-owned apps.* Retrieved November 11, 2024 from https://developer.android.com/training/articles/app-set-id
[49] Android Developers. 2024. *Meet Google Play's target API level requirement.* Retrieved November 11, 2024 from https://developer.android.com/google/play/requirements/target-sdk
[50] Constanze Dietrich, Katharina Krombholz, Kevin Borgolte, and Tobias Fiebig. 2018. Investigating system operators' perspective on security misconfigurations. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. 1272–1289.
[51] Chartboost Documentation. 2024. *SDK Privacy Methods.* Retrieved November 11, 2024 from https://docs.chartboost.com/en/monetization/integrate/android/sdk-privacy-methods/
[52] Unity Documentation. 2023. Consumer privacy act compliance. Retrieved November 11, 2024 from https://docs.unity.com/ads/en-us/manual/CCPACompliance
[53] Unity Documentation. 2023. Google Play data safety section for Unity Ads. Retrieved November 11, 2024 from https://docs.unity.com/ads/en-us/manual/GoogleDataSafety
[54] Unity Documentation. 2024. *Installing the Unity Ads SDK for Android.* Retrieved November 11, 2024 from https://docs.unity.com/ads/en-us/manual/InstallingTheAndroidSDK
[55] Manuel Egele, David Brumley, Yanick Fratantonio, and Christopher Kruegel. 2013. An empirical study of cryptographic misuse in android applications. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 73–84.
[56] Anirudh Ekambaranathan, Jun Zhao, and Max Van Kleek. 2021. "Money makes the world go around": Identifying Barriers to Better Privacy in Children's Apps From Developers' Perspectives. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–15.
[57] Sascha Fahl, Marian Harbach, Thomas Muders, Lars Baumgärtner, Bernd Freisleben, and Matthew Smith. 2012. Why Eve and Mallory love Android: An Analysis of Android SSL (in) Security. In *Proceedings of the 2012 ACM conference on Computer and communications security*. 50–61.
[58] Álvaro Feal, Paolo Calciati, Narseo Vallina-Rodriguez, Carmela Troncoso, Alessandra Gorla, et al. 2020. Angel or devil? a privacy study of mobile parental control apps. *Proceedings of Privacy Enhancing Technologies (PoPETS)* 2020 (2020).
[59] Álvaro Feal, Julien Gamba, Juan Tapiador, Primal Wijesekera, Joel Reardon, Serge Egelman, and Narseo Vallina-Rodriguez. 2021. Don't accept candy from strangers: An analysis of third-party mobile sdks. *Data Protection and Privacy, Volume 13: Data Protection and Artificial Intelligence* 13 (2021), 1.
[60] Adrienne Porter Felt, Erika Chin, Steve Hanna, Dawn Song, and David Wagner. 2011. Android permissions demystified. In *Proceedings of the 18th ACM conference on Computer and communications security*. 627–638.
[61] Google Firebase. 2024. . Retrieved November 16, 2024 from https://firebase.google.com/
[62] Flurry. 2023. Flurry Analytics Terms of Service. Retrieved January 6, 2024 from https://developer.yahoo.com/flurry/legal-privacy/terms-service/flurry-analytics-terms-service.html
[63] Flurry. 2024. Flurry Analytics. Retrieved November 11, 2024 from https://www.flurry.com/
[64] FTC 2024. Federal Trade Commission (FTC). Retrieved November 11, 2024 from https://www.ftc.gov/
[65] Federal Trade Commission (FTC). 2016. InMobi FTC case. Retrieved November 11, 2024 from https://www.ftc.gov/legal-library/browse/cases-proceedings/152-3203-inmobi-pte-ltd
[66] Federal Trade Commission (FTC). 2020. Hyberbeard FTC case. Retrieved November 11, 2024 from https://www.ftc.gov/legal-library/browse/cases-proceedings/192-3109-hyperbeard-inc
[67] Federal Trade Commission (FTC). 2021. Kuuhuub FTC case. Retrieved November 11, 2024 from https://www.ftc.gov/legal-library/browse/cases-proceedings/182-3184-kuuhuub-inc-et-al-us-v-recolor-oy
[68] Conor Gilsenan, Fuzail Shakir, Noura Alomar, and Serge Egelman. 2023. Security and Privacy Failures in Popular 2FA Apps. In *32nd USENIX Security Symposium (USENIX Security 23)*.
[69] Aniketh Girish, Tianrui Hu, Vijay Prakash, Daniel J Dubois, Srdjan Matic, Danny Yuxing Huang, Serge Egelman, Joel Reardon, Juan Tapiador, David Choffnes, et al. 2023. In the Room Where It Happens: Characterizing Local Communication and Threats in Smart Homes. In *Proceedings of the 2023 ACM on Internet Measurement Conference*. 437–456.
[70] Catherine Han, Irwin Reyes, Álvaro Feal, Joel Reardon, Primal Wijesekera, Narseo Vallina-Rodriguez, Amit Elazar, Kenneth A Bamberger, and Serge Egelman. 2020. The price is (not) right: Comparing privacy in free and paid apps. *Proceedings on Privacy Enhancing Technologies* 2020, 3 (2020).
[71] Google Play Console Help. 2023. *Advertising ID.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/6048248
[72] Google Play Console Help. 2023. *Best practices for prominent disclosure and consent.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/11150561
[73] Google Play Console Help. 2023. *Data practices in Families apps.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/11043825
[74] Google Play Console Help. 2023. Prepare your app for review. Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/9859455
[75] Google Play Console Help. 2024. *Manage target audience and app content settings.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/9867159
[76] Google Play Console Help. 2024. *Target API level requirements for Google Play apps.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/11926878
[77] Google Play Policy Help. 2024. *Participate in the Families Self-Certified Ads SDK Program.* Retrieved November 11, 2024 from https://support.google.com/googleplay/android-developer/answer/12955712
[78] InMobi. 2024. InMobi Services. Retrieved November 11, 2024 from https://www.inmobi.com/
[79] intersoft consulting. 2016. General Data Protection Regulation (GDPR). Retrieved November 11, 2024 from https://gdpr-info.eu/
[80] IPWHOIS.IO 2024. IP Geolocation API. Retrieved November 11, 2024 from https://ipwhois.io/
[81] ironSource. 2023. ironSource - Mediation Networks for Android. Retrieved January 3, 2024 from https://developers.is.com/ironsource-mobile/android/mediation-networks-android/
[82] ironSource. 2023. IronSource - Regulation Advanced Settings. Retrieved September 26, 2023 from https://developers.is.com/ironsource-mobile/android/regulation-advanced-settings/#step-1
[83] ironSource. 2023. ironSource: Google's Data Safety Questionnaire. Retrieved November 30, 2023 from https://developers.is.com/ironsource-mobile/general/googles-data-safety-questionnaire-full/
[84] ironSource. 2023. ironSource Mobile Ltd. Privacy Policy. Retrieved November 30, 2023 from https://developers.ironsrc.com/ironsource-mobile/air/ironsource-mobile-privacy-policy
[85] ironSource. 2024. IronSource Services. Retrieved November 11, 2024 from https://www.is.com/
[86] Rishabh Khandelwal, Asmit Nayak, Paul Chung, and Kassem Fawaz. 2023. Unpacking Privacy Labels: A Measurement and Developer Perspective on Google's Data Safety Section. *arXiv preprint arXiv:2306.08111* (2023).
[87] Simon Koch, Benjamin Altpeter, and Martin Johns. 2023. The {OK} Is Not Enough: A Large Scale Study of Consent Dialogs in Smartphone Applications. In *32nd USENIX Security Symposium (USENIX Security 23)*. 5467–5484.
[88] Simon Koch, Malte Wessels, Benjamin Altpeter, Madita Olvermann, and Martin Johns. 2022. Keeping privacy labels honest. *Proceedings on Privacy Enhancing Technologies* 4, 486-506 (2022), 2–2.
[89] Konrad Kollnig, Pierre Dewitte, Max Van Kleek, Ge Wang, Daniel Omeiza, Helena Webb, and Nigel Shadbolt. 2021. A Fait Accompli? An Empirical Study into the Absence of Consent to {Third-Party} Tracking in Android Apps. In *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. 181–196.
[90] Konrad Kollnig, Anastasia Shuba, Reuben Binns, Max Van Kleek, and Nigel Shadbolt. 2022. Are iphones really better for privacy? comparative study of ios and android apps. *The 22nd Privacy Enhancing Technologies Symposium (PETS 20122))* (2022).
[91] Konrad Kollnig, Anastasia Shuba, Max Van Kleek, Reuben Binns, and Nigel Shadbolt. 2022. Goodbye tracking? Impact of iOS app tracking transparency and privacy labels. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*. 508–520.
[92] Tianshi Li, Kayla Reiman, Yuvraj Agarwal, Lorrie Faith Cranor, and Jason I Hong. 2022. Understanding challenges for developers to create accurate privacy nutrition labels. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*. 1–24.
[93] Allan Lyons, Julien Gamba, Austin Shawaga, Joel Reardon, Juan Tapiador, Serge Egelman, Narseo Vallina-Rodriguez, et al. 2023. Log: It's Big, It's Heavy, It's Filled with Personal Data! Measuring the Logging of Sensitive Information in the Android Ecosystem. In *Usenix Security Symposium*.
[94] Tinhinane Medjkoune, Oana Goga, and Juliette Senechal. 2023. Marketing to Children Through Online Targeted Advertising: Targeting Mechanisms and Legal Aspects. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 180–194.
[95] Meta. 2024. Information for Child-Directed Apps and Services. Retrieved November 11, 2024 from https://developers.facebook.com/docs/audience-network/optimization/best-practices/coppa/

[96] Meta. 2024. Meta App Events SDK. Retrieved November 11, 2024 from https://developers.facebook.com/docs/app-events/getting-started-app-events-android

[97] Meta. 2024. Meta Audience Network SDK. Retrieved November 11, 2024 from https://developers.facebook.com/docs/audience-network/setting-up/ad-setup/android/

[98] Abraham H Mhaidli, Yixin Zou, and Florian Schaub. 2019. " We Can't Live Without {Them!}" App Developers' Adoption of Ad Networks and Their Considerations of Consumer Risks. In *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. 225–244.

[99] Meta Audience Network. 2023. Add the Audience Network SDK to your Android App. Retrieved September 25, 2023 from https://developers.facebook.com/docs/audience-network/setting-up/platform-setup/android/add-sdk

[100] Trung Tin Nguyen, Michael Backes, Ninja Marnau, and Ben Stock. 2021. Share First, Ask Later (or Never?) Studying Violations of GDPR's Explicit Consent in Android Apps. In *30th USENIX Security Symposium (USENIX Security 21)*. 3667–3684.

[101] Trung Tin Nguyen, Michael Backes, and Ben Stock. 2022. Freely given consent? Studying consent notice of third-party tracking and its violations of GDPR in Android apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 2369–2383.

[102] Office of the Attorney General. 2024. California Consumer Privacy Act (CCPA). Retrieved November 11, 2024 from https://oag.ca.gov/privacy/ccpa

[103] Information Commissioner's Office. 2016. GDPR - Data minimisation. Retrieved November 11, 2024 from https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/childrens-information/childrens-code-guidance-and-resources/age-appropriate-design-a-code-of-practice-for-online-services/8-data-minimisation/

[104] Ehimare Okoyomon, Nikita Samarin, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, Irwin Reyes, Álvaro Feal, Serge Egelman, et al. 2019. On the ridiculousness of notice and consent: Contradictions in app privacy policies. In *Workshop on Technology and Consumer Protection (ConPro 2019), in conjunction with the 39th IEEE Symposium on Security and Privacy*.

[105] Marten Oltrogge, Nicolas Huaman, Sabrina Amft, Yasemin Acar, Michael Backes, and Sascha Fahl. 2021. Why Eve and Mallory Still Love Android: Revisiting {TLS}({In) Security} in Android Applications. In *30th USENIX Security Symposium (USENIX Security 21)*. 4347–4364.

[106] OneSignal. 2024. OneSignal Services. Retrieved November 11, 2024 from https://onesignal.com/

[107] Google Play. 2023. *Build Teacher Approved apps*. Retrieved July 10, 2023 from https://play.google.com/console/about/programs/teacherapproved/

[108] Google Play. 2024. *Developer Console*. Retrieved November 11, 2024 from https://play.google.com/console/about/

[109] Google Play. 2024. *Developer Policy Center*. Retrieved November 11, 2024 from https://play.google.com/about/developer-content-policy/

[110] Google Play. 2024. Policy announcement: April 5, 2023. Retrieved November 14, 2024 from https://support.google.com/googleplay/android-developer/answer/13411745?sjid=14219206286122674930-NC

[111] Google Play. 2024. Policy announcement: November 16, 2022. Retrieved November 14, 2024 from https://support.google.com/googleplay/android-developer/answer/14550832?sjid=14219206286122674930-NC

[112] Google Play. 2024. Policy Archive. Retrieved November 14, 2024 from https://support.google.com/googleplay/android-developer/answer/13386702?sjid=14219206286122674930-NC

[113] Ole André V. Ravnås. 2023. Frida. Retrieved November 11, 2024 from https://frida.re/docs/android/

[114] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, Phillipa Gill, et al. 2018. Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem. In *The 25th Annual Network and Distributed System Security Symposium (NDSS 2018)*.

[115] Joel Reardon, Álvaro Feal, Primal Wijesekera, Amit Elazari Bar On, Narseo Vallina-Rodriguez, and Serge Egelman. 2019. 50 ways to leak your data: An exploration of apps' circumvention of the Android permissions system. In *28th USENIX security symposium (USENIX security 19)*. 603–620.

[116] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, Serge Egelman, et al. 2018. "Won't Somebody Think of the Children?" Examining COPPA Compliance at Scale. In *The 18th Privacy Enhancing Technologies Symposium (PETS 2018)*.

[117] Nikita Samarin, Shayna Kothari, Zaina Siyed, Oscar Bjorkman, Reena Yuan, Primal Wijesekera, Noura Alomar, Jordan Fischer, Chris Hoofnagle, and Serge Egelman. 2023. Lessons in VCR Repair: Compliance of Android App Developers with the California Consumer Privacy Act (CCPA). *Proceedings on Privacy Enhancing Technologies* (2023).

[118] Google Play services. 2024. *AdvertisingIdClient.Info*. Retrieved November 11, 2024 from https://developers.google.com/android/reference/com/google/android/gms/ads/identifier/AdvertisingIdClient.Info

[119] Kochava Services. 2024. Kochava. Retrieved November 11, 2024 from https://www.kochava.com/

[120] Singular. 2024. Singular Services. Retrieved November 11, 2024 from https://www.singular.net/

[121] StackOverflow. 2022. *Issue found: Invalid Data safety section. How to fix this issue? [closed]*. Retrieved November 16, 2024 from https://stackoverflow.com/questions/71217909/issue-found-invalid-data-safety-section-how-to-fix-this-issue

[122] StackOverflow. 2022. *Your app's Data safety section will be invalidated and state No information available*. Retrieved November 16, 2024 from https://stackoverflow.com/questions/73849730/your-apps-data-safety-section-will-be-invalidated-and-state-no-information-ava

[123] Start.io. 2022. Google Play Safety Section - Data Disclosure. Retrieved November 11, 2024 from https://support.start.io/hc/en-us/articles/4413793394962-Google-Play-Safety-Section-Data-Disclosure

[124] Start.io. 2024. Android SDK. Retrieved November 11, 2024 from https://support.start.io/hc/en-us/articles/360014774799-Integration-via-Maven#addingsdktoyourproject

[125] statcounter. 2024. Android Version Market Share Worldwide. Retrieved January 21, 2024 from https://gs.statcounter.com/os-version-market-share/android

[126] Ruoxi Sun, Minhui Xue, Gareth Tyson, Shuo Wang, Seyit Camtepe, and Surya Nepal. 2023. Not Seen, Not Heard in the Digital World! Measuring Privacy Practices in Children's Apps. In *Proceedings of the ACM Web Conference 2023*. 2166–2177.

[127] Unity Technologies. 2024. Unity Ads: Mobile Game Ad Network Platform & Analytics. Retrieved November 11, 2024 from https://unity.com/products/unity-ads

[128] Umeng. 2024. Umeng Services. Retrieved November 11, 2024 from https://www.umeng.com/

[129] Christine Utz, Martin Degeling, Sascha Fahl, Florian Schaub, and Thorsten Holz. 2019. (Un) Informed Consent: Studying GDPR Consent Notices in the Field. In *Proceedings of the 2019 acm sigsac conference on computer and communications security*. 973–990.

[130] Vungle. 2023. Vungle - Advanced Settings. Retrieved January 3, 2024 from https://support.vungle.com/hc/en-us/articles/360047780372-Advanced-Settings

[131] Vungle. 2024. Vungle Services. Retrieved November 11, 2024 from https://vungle.com

[132] Zack Whittaker. 2019. *Apple restricts ads and third-party trackers in iPhone apps for kids*. Retrieved November 11, 2024 from https://techcrunch.com/2019/06/03/apple-kid-apps-trackers/

[133] Anhao Xiang, Weiping Pei, and Chuan Yue. 2023. PolicyChecker: Analyzing the GDPR Completeness of Mobile Apps' Privacy Policies. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 3373–3387.

[134] Yue Xiao, Zhengyi Li, Yue Qin, Xiaolong Bai, Jiale Guan, Xiaojing Liao, and Luyi Xing. 2023. Lalaine: Measuring and Characterizing {Non-Compliance} of Apple Privacy Labels. In *32nd USENIX Security Symposium (USENIX Security 23)*. 1091–1108.

[135] Yandex. 2023. Yandex Mobile Ads SDK for Android. Retrieved November 11, 2024 from https://yandex.ru/support2/mobile-ads/en/dev/android

[136] Yuqing Yang, Mohamed Elsabagh, Chaoshun Zuo, Ryan Johnson, Angelos Stavrou, and Zhiqiang Lin. 2022. Detecting and Measuring Misconfigured Manifests in Android Apps. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. 3063–3077.

[137] Eric Zeng, Frank Li, Emily Stark, Adrienne Porter Felt, and Parisa Tabriz. 2019. Fixing HTTPS misconfigurations at scale: An experiment with security notifications. *Workshop on the Economics of Information Security (WEIS)* (2019).

[138] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven Bellovin, and Joel Reidenberg. 2016. Automated analysis of privacy requirements for mobile apps. In *2016 AAAI Fall Symposium Series*.

A Developer survey

This section lists the questions included in our developer survey which helped us evaluate the extent to which configuring third-party SDKs for child-appropriate treatment and preparing accurate privacy labels are supported within existing app development processes. The survey also asked developers about the types of privacy guidance that they relied on and their perspectives on how such guidance can be improved.

A.1 Part 1: Third-party Software Development Kits (SDKs)

(1) Which of the following third-party SDK providers receive personal data from the child-directed apps that you published on the Google Play Store?
(2) For what purposes does your organization use third-party SDKs in its mobile apps?
(3) Many third-party SDKs offer privacy configurations that can be used to signal to an SDK that children are among an app's target audiences, that a user has provided consent (or not) or to limit the collection of certain data types (e.g., advertising IDs and location data). Are you familiar with such types of third-party SDK privacy configurations?
(4) To what extent does your organization make use of third-party SDK privacy compliance configurations/settings to limit data sharing with third parties?
(5) What are the potential consequences to your organization for not configuring third-party SDK privacy compliance settings correctly?
(6) To what extent does your organization find that it is challenging to correctly use third-party SDK privacy compliance configurations to comply with applicable privacy regulations?
(7) What kind of challenges did your team experience in their effort to configure third-party SDKs for compliance with privacy regulations?
(8) Does your organization dedicate resources to ensuring that third-party SDKs bundled within your organization's mobile apps are kept up-to-date?
(9) To what extent does your organization trust that personal data collected by third-party SDKs through your app(s) will not be used for purposes that are prohibited (e.g., behavioral advertising) under applicable privacy laws or the Google Play Store policies?
(10) What kind of improvements/changes would your organization like third-party SDK providers to make in order to make it easier for your organization to use third-party SDK privacy compliance configurations correctly?

A.2 Part 2: Data Safety labels

(1) Who is responsible for filling out the details for your apps' data safety labels?
(2) How would you rate the difficulty of accurately filling out the details required for data safety labels on the Google Play store?
(3) Does your development process dedicate time and/or resources to ensuring the accuracy of information included in your apps' data safety labels?
(4) If Yes, what kind of process did your team(s) follow to ensure the accuracy of the information included in your apps' data safety labels?
(5) How does your team(s) identify the types of data collected by third-party SDKs before filling out your apps' data safety labels?
(6) Do you use any dynamic or static testing tools that allow you to identify the types of data collected by your app(s) before filling out the details needed for data safety labels (e.g., tools for capturing network traffic)?
(7) If Yes, could you provide more details on the tools that you use for testing your app(s) to identify the types of data your app(s) collect and/or share?
(8) What kind of challenges did your team(s) face while filling out the details required for your apps' data safety labels?
(9) If you were to make changes to the Google Play developer console to make the process of filling out data safety labels easier for app developers, what would you do?
(10) Should filling out the details of the data safety label be automatically done by Google Play instead of leaving this task to app developers?
(11) Do you believe that Google checks your app(s)' data safety label(s) for accuracy?
(12) In your opinion, what are the potential consequences for having inaccuracies in the disclosures made in apps' data safety labels?

A.3 Part 3: Privacy guidance for app developers

(1) What kind of resources do you have access to through your organization for guidance on how to improve the privacy of your app(s) or fulfill your privacy compliance obligations?
(2) Does your developer organization currently provide you with specific guidance or resources to help you improve the privacy of your app(s) or comply with applicable privacy regulations?
(3) If Yes, what kind of privacy guidance do you currently have access to?
(4) If you were to be offered privacy guidance, would you use it in your development tasks?
(5) If Yes, what type of privacy guidance would you most want to have?
(6) Where, in your development process, do you think that developer privacy guidance would be most likely to be considered?
(7) If you were to be offered a specific type of privacy developer guidance, in what format would you like to receive this guidance?
(8) Where would you like to see the privacy developer guidance presented to you?
(9) What kind of content in third-party SDK documentation do you find challenging to understand or find?
(10) What kind of help do you need to understand the data collection behaviors of third-party SDKs integrated in your app?
(11) If you were to be offered a specific type of documented privacy developer guidance, what type of technical content would you like this guidance to have?
(12) If you were to be offered a specific type of documented privacy developer guidance, what type of legal content would you like this guidance to have?

(13) Assuming that we provided you with documented guidance to help you improve the privacy of your app(s), what else would you need?

B Declared permissions

Table 4 demonstrates the number of child-directed apps that declared permissions which allow accessing children's personal data.

Table 4: Number of apps that declared Android permissions.

Permission	# of apps (N=7,377)
READ_EXTERNAL_STORAGE	2793 apps (38%)
AD_ID	1901 apps (26%)
CAMERA	688 apps (9.3%)
READ_PHONE_STATE	605 apps (8.2%)
ACCESS_FINE_LOCATION	160 apps (2.2%)
ACCESS_COARSE_LOCATION	152 apps (2.06%)
READ_MEDIA_IMAGES	122 apps (1.7%)
GET_ACCOUNTS	116 apps (1.6%)
READ_MEDIA_VIDEO	103 apps (1.4%)
READ_MEDIA_AUDIO	94 apps (1.3%)
READ_CONTACTS	34 apps (0.46%)
ACCESS_MEDIA_LOCATION	4 apps (0.05%)
ACCESS_BACKGROUND_LOCATION	2 apps (0.03%)
BODY_SENSORS	2 apps (0.03%)
READ_PHONE_NUMBERS	2 apps (0.03%)
READ_SMS	0 apps (0%)
BODY_SENSORS_BACKGROUND	0 apps (0%)

C Traffic snapshots

```
POST /configure HTTP/1.1
User-Agent: Mozilla/5.0 (Linux; Android 12; AOSP on sargo
Build/SP2A.220505.008; wv) AppleWebKit/537.36 (KHTML, like Gecko)
Version/4.0 Chrome/112.0.5597.0 Mobile Safari/537.36
Content-Type: application/octet-stream
Req-Dict-Id: conf-req-001
Resp-Dict-Id: conf-res-001
Content-Length: 416
Host: androidads4-8.adcolony.com
Connection: Keep-Alive
Accept-Encoding: gzip
```

```
{"origin_store":"google","coppa_required":true,"mediation_network":"
AdMob","mediation_network_version":"4.8.0.1","app_id":"*****","bundl
e_id":<packagename ->
"*****">,"os_name":"android","advertiser_id":"61064d92-87ce-4a6f-
b309-
449cd2146924","carrier":"unknown","custom_id":"","limit_tracking":fa
lse,"ln":"en","locale":"US","device_brand":"Google","device_model":"
AOSP on
sargo","device_type":"phone","network_type":"wifi","os_version":"12"
,"sdk_version":"4.8.0","battery_level":1,"sdk_type":"android_native"
,"current_orientation":1,"cell_service_country_code":"","bundle_vers
ion_short":"4.0.8","adc_alt_id":"ceea793d-f19e-45e1-96e6-
8dc731662245","app_set_id":"530f6f11-8567-56a5-f840-
50ebecc39afd","ad_history":{},"ad_playing":{},"ad_queue":{},"sid":"3
d6b8422-b0a9-4e94-b651-
38ddb19dc05e","device_time":1685032554378,"real_controller_version":
"3.3.2","controller_version":"3.3.2","ad_request":false,"screen_heig
ht":1080,"screen_width":2088,"available_stores":["google"],"manufact
urer":"Google","cleartext_permitted":false,"device_audio":false}
```

Figure 4: Collection of AAIDs and app set IDs alongside each other despite correct use of AdColony's [1] COPPA configuration.

```
POST /mediation?adUnit=3&sessionId=95da3537-a4be-490d-a386-
54eca217aeac&appKey=*****&compression=true HTTP/1.1
Content-Type: application/json
Content-Length: 912
User-Agent: Dalvik/2.1.0 (Linux; U; Android 12; AOSP on sargo
Build/SP2A.220505.008)
Host: outcome-ssp.supersonicads.com
Connection: Keep-Alive
Accept-Encoding: gzip
```

```
{"userIdType":"GAID","userId":"95da3537-a4be-490d-a386-
54eca217aeac","appKey":"*****","isLimitAdTrackingEnabled":false,"app
Version":"1.2.3","tz":"America\/Los_Angeles","icc":"us","advertising
Id":"95da3537-a4be-490d-a386-
54eca217aeac","language":"en","battery":100,"mcc":0,"connectionType"
:"wifi","internalFreeMemory":45835,"osVersion":"32(12)","firstSessio
n":"false","deviceOEM":"Google","auid":"7365b13d-b33c-4c12-b47d-
931693dc3b04","mnc":0,"deviceOS":"Android","mt":"AdMob310","bundleId
":<packagename -> "*****">,"sessionId":"d80045e9-0066-4b32-8cd5-
122b648f99b2","externalFreeMemory":45835,"advertisingIdType":"GAID",
"jb":"false","sdkVersion":"7.2.7","deviceModel":"AOSP on
sargo","gmtMinutesOffset":-
420,"abt":"A","internalTestId":"{}","is_coppa":"true","controllerABV
alue":"11","asid":"ffb3a997-08e3-20d9-9a38-
328529c27a2c","timestamp":1684786670992,"events":[{"provider":"Media
tion","isDemandOnly":1,"interstitial":true,"rewardedVideo":true,"ext
1":"appLanguage=Kotlin,kiag,androidx=true,Activity=true,cachedUserAg
ent=false","sessionDepth":1,"eventSessionId":"d80045e9-0066-4b32-
8cd5-
122b648f99b2","connectionType":"wifi","firstSessionTimestamp":168478
6347542,"eventId":14,"timestamp":1684786670192}]}
```

Figure 5: Collection of AAIDs and app set IDs despite correct use of ironSource's [82] COPPA configuration.

D Privacy Configurations: Measurement of Adoption

Tables 5 and 6 provide detailed measurements of the use of third-party SDK privacy configurations in the tested child-directed apps that were available on the Google Play Store.

Table 5: Adoption of COPPA-related third-party SDK privacy configuration for recent app versions (flag value* denotes a privacy-friendly configuration).

Privacy flag	Third-party SDK	Receiving domain(s)	# of apps that contacted the domain(s)	# apps per each flag value	# of apps that shared AAIDs [correctly configured, incorrectly configured, configuration not used]
COPPA	Meta Audience Network [97]	facebook.com	339	TRUE*: 0, FALSE: 24	[0, 20, 0]
is_coppa				TRUE*: 281, FALSE: 9	[41, 9, 6]
is_child_directed				TRUE*: 227, FALSE: 4	[6, 4, 46]
google_family_self_certified_sdks				TRUE*: 0, FALSE: 0	[0, 0, 56]
AdColony_COPPA				TRUE*: 144, FALSE: 0	[4, 0, 52]
AdColony_APP_Child_Directed				TRUE*: 139, FALSE: 0	[0, 0, 56]
AdMob_TFCD				TRUE*: 176, FALSE: 0	[4, 0, 52]
AdMob_TFUA	ironSource [85]	supersonicads.com	299	TRUE*: 116, FALSE: 0	[4, 0, 52]
Chartboost_Coppa				TRUE*: 20, FALSE: 0	[0, 0, 56]
Pangle_COPPA				TRUE or 1*: 70, FALSE: 0	[0, 0, 56]
Vungle_coppa				TRUE or 1*: 178, FALSE: 0	[0, 0, 56]
InMobi_AgeRestricted				TRUE*: 171, FALSE: 0	[0, 0, 56]
AppLovin_AgeRestrictedUser				TRUE*: 61, FALSE: 0	[4, 0, 52]
META_Mixed_Audience				TRUE*: 55 , FALSE: 0	[0, 0, 56]
coppa				TRUE*: 430, FALSE: 35	[9, 33, 87]
coppaCompliant				TRUE*: 453, FALSE: 38	[20, 36, 73]
appLevelCoppa				TRUE*: 455, FALSE: 37	[20, 36, 73]
calculatedCoppa	Unity [54]	unity3d.com	741	TRUE*: 454, FALSE: 38	[20, 36, 73]
user.nonBehavioral				TRUE*: 13, FALSE: 14	[0, 14, 115]
contextualOnly				TRUE*: 0, FALSE: 492	[0, 56, 73]
coppa	Appodeal [12]	appbaqend.com	4	TRUE*: 0, FALSE: 4	[0, 1, 0]
for_kids				TRUE*: 0, FALSE: 4	[0, 1, 0]
is_coppa	Vungle [131]	vungle.com	203	TRUE*: 167, FALSE: 8	[0, 5, 7]
age_restricted_user	Yandex [135]	yandex.net, yandex.ru	40	0: 0, 1*: 0	[0, 0, 16]
coppa				TRUE or 1*: 110, FALSE: 44	[16, 6, 5]
applyGdprAgeOfConsent	InMobi [78]	inmobi.com	169	TRUE*: 108 FALSE: 45	[16, 6, 5]
u-age-restricted				1*: 112, 0: 7	[1, 6, 20]
coppa	Chartboost [31]	chartboost.com	91	TRUE*: 39, FALSE: 0	[0, 0, 49]
coppa_required	AdColony [2]	adcolony.com	158	TRUE*: 115, FALSE: 7	[5, 5, 24]
is_child_directed				TRUE*: 109, FALSE: 0	[0, 0, 34]
coppa				TRUE or 1*: 145, FALSE or 0: 10	[25, 8, 1]
tag_for_child_directed_treatment or tfcd	Google AdMob [3]	doubleclick.net, fundingchoicesmessages.google.com	3147	NaN: 44, 1*: 1159, 0: 43	[19, 2, 43]
tag_for_under_age_of_consent or tfua				TRUE or 1*: 390, FALSE or 0: 41, NaN: 125	[32, 28, 4]
is_age_restricted_user	AppLovin [11]	applovin.com	163	TRUE*: 32, FALSE: 4	[6, 3, 40]

Table 6: Adoption of CCPA and GDPR third-party SDK privacy configurations, and configurations that can disable collecting personal data for recent app versions. (flag value* denotes a privacy-friendly configuration)

Privacy flag	Third-party SDK	Receiving domain(s)	# of apps that contacted the domain(s)	# of apps per each flag value	# of apps that shared AAIDs [correctly configured, incorrectly configured, configuration not used]
advertiser_id_collection_enabled	Meta's event tracking SDK [96]	facebook.com	339	TRUE: 96, FALSE*: 66	[0, 87, 113]
DATA_PROCESSING_OPTIONS	Meta Audience Network [97]	facebook.com	339	null: 21, LDU*: 0	[0, 0, 20]
DATA_PROCESSING_OPTIONS_COUNTRY				null: 21, 1: 0	[0, 0, 20]
DATA_PROCESSING_OPTIONS_COUNTRY_STATE				null: 21, 1000: 0	[0, 0, 20]
do_not_sell	ironSource [85]	supersonicads.com	299	TRUE*: 205, FALSE: 4	[7, 4, 45]
metadata_consent				FALSE or 0: 102, TRUE or 1: 6	[6, 4, 46]
is_deviceid_optout				TRUE*: 224, FALSE: 0	[4, 0, 52]
gdpr.consent	Unity [54]	unity3d.com	741	TRUE: 9, FALSE: 86	[4, 3, 122]
privacy.consent				TRUE: 16, FALSE*: 163	[1, 8, 120]
disable_ad_id	Vungle [131]	vungle.com	203	TRUE*: 177, FALSE: 11	[0, 10, 2]
user.ccpa.status				opted_in: 49, opted_out*: 149	[0, 12, 0]
user.gdpr.consent_status				opted_in: 4, opted_out: 57, unknown: 142	[1, 3, 8]
user_consent	Yandex [135]	yandex.net, yandex.ru	40	0: none, 1: 1	[0, 1, 15]
fl.ccpa.optout	Flurry [63]	flurry.com	34	TRUE*: 0 , FALSE: 10	[0, 0, 14]
fl.report.location.enabled				TRUE: 7, FALSE*: 5	[NA, NA, NA]
gdpr_consent_available	InMobi [78]	inmobi.com	169	TRUE: 2, FALSE: 43	[4, 2, 21]
do_not_sell				1: 57, 0: 0	[0, 0, 27]
pidatauseconsent	Chartboost [31]	chartboost.com	91	0: 4, -1: 74 , 1: 4	[2, 4, 43]
us_privacy				1YY-*: 21 , 1NY-*: 1 , 1NN-: 3, 1YN-: 2 , NULL: 26	[1, 3, 45]
gdpr_required	AdColony [2]	adcolony.com	158	TRUE: 28, FALSE: 58	[3, 6, 25]
gdpr_consent_string				1: 5 , 0: 25	[3, 3, 28]
ccpa_required				TRUE*: 117, FALSE: 2	[8, 0, 26]
ccpa_consent_string				1: 6 , 0: 113	[4, 4, 26]
npa	Google AdMob [3]	doubleclick.net, fundingchoicesmessages.google.com	3147	1*: 342, 0: 27	[6, 2, 56]
rdp				1*: 188, 0: 13	[0, 0, 64]
has_user_consent	AppLovin [11]	applovin.com	163	TRUE: 11, FALSE: 13	[4, 3, 42]
is_do_not_sell				TRUE*: 26, FALSE: 2	[3, 0, 46]

E Linking of Different Data Types

Figure 6 shows the total number of apps that sent AAIDs alongside other types of personal data, which allows bridging AAIDs after they have been reset by users. Most of the linking was due to transmitting FIDs alongside AAIDs to app-measurement.com, which is owned by Firebase [61].

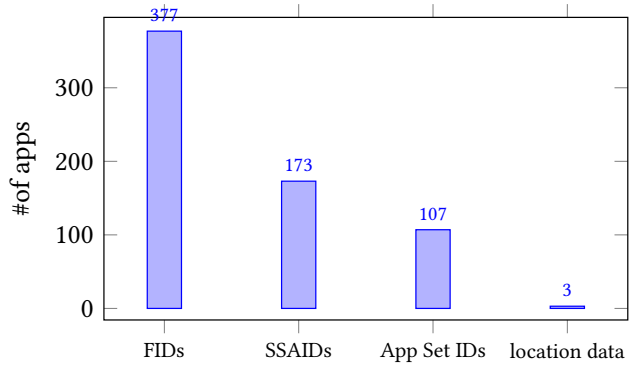

Figure 6: Number of apps that transmitted AAIDs alongside other types of data to app-measurement.com and other domains.

F Example of a Data Safety Label

Figure 7 shows an example of a data safety label prepared by a developer for one of their DFF-badged apps (5). In this example, the developer disclosed that their app shares two of the data types defined in Google Play's specification for data safety labels [22], which are "device IDs" and "app info and performance". The developer also indicated that their app does not collect personal data (2), that encryption is used in their communications that include the disclosed data types (3) and that users cannot request the deletion of their data (4). In our analyses, we examined the accuracy of these disclosures by comparing them to apps' actual data collection and sharing behaviors. For example, if we observed the app whose label is included in Figure 7 transmitted AAIDs to *live.chartboost.com*, we considered that an accurate disclosure.

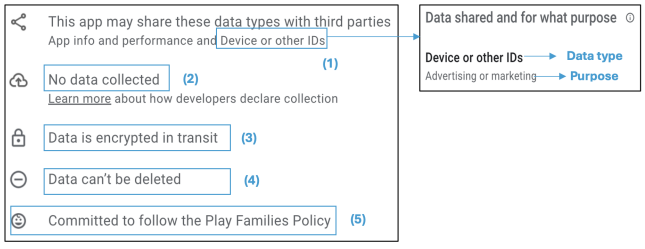

Figure 7: Example of a data safety label.

Figure 8 shows an example of data safety label whose developer disclosed that their app does not collect (1) or share (2) any of the data types considered in Google Play's specification for data safety labels [22]. In this example, if we observed that the app whose label is included in Figure 8 transmitted AAIDs to *graph.facebook.com*, we considered that an inaccurate disclosure of the transmitted data type and the purpose of such transmission.

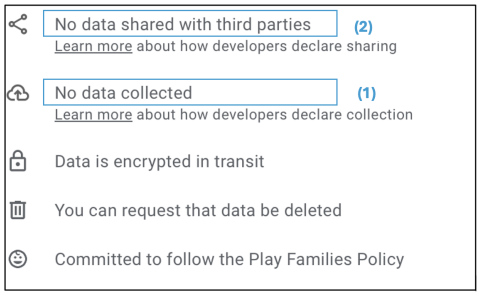

Figure 8: Example of a data safety label disclosing that no data is collected or shared.

Figure 9 depicts an app that did not have a data safety label available on the store at the time of testing. This could result from developers not preparing a label for their apps or preparing one that the store removed after finding inaccurate disclosures [122].

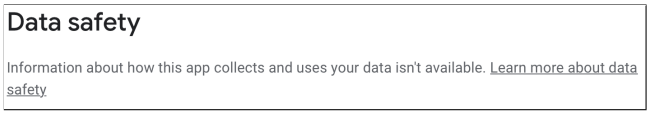

Figure 9: Example of an app that did not have a data safety label.

G Timeline of policy changes

Figure 10 provides an timeline that demonstrates when relevant Google Play policies were introduced or became effective. This timeline is based on the details available on Google Play's policy archive [112]. The dates shown in Figure 10 might not necessarily demonstrate whether these policies were enforced by Google Play at the time. However, anecdotal evidence from developer forums could provide evidence about the dates on which Google Play started enforcing its policies. For example, developers posted questions in late 2022 on StackOverflow (e.g., [121, 122]) that requested guidance on how to fix their data safety labels to react to notifications they received from Google Play.

To our knowledge, all the policies relevant to our experiments were effective before we started our experiments (early February 2023) except for two of them. One of which required developers to update the API levels targeted by their apps and the other one required using specific versions for self-certified SDKs in child-directed apps. The ongoing enforcement of Google Play's target API level policy that was happening during or after our pilot tests could explain why we were only able to test 4,975 apps in the second app runs (Section 4.2). While developers of child-directed apps were required to comply with the SDK versions policy starting from May 31 2023 [110], the policy was announced in late November 2022 which could have led developers to update SDKs to one of their more recent versions [111].

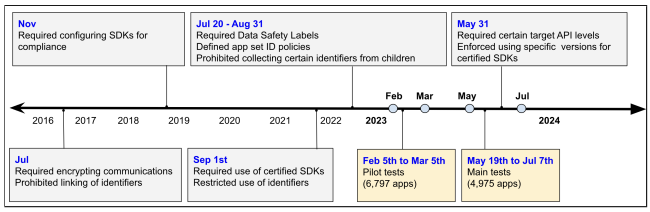

Figure 10: Timeline showing when relevant Google Play's policies were introduced.