Uncovering Relationships between Android Developers, User Privacy, and Developer Willingness to Reduce Fingerprinting Risks

Alex Berke
aberke@google.com
Google
Cambridge, MA, USA

Güliz Seray Tuncay
gulizseray@google.com
Google
San Francisco, CA, USA

Michael Specter
mikespecter@google.com
Google
Atlanta, GA, USA

Mihai Christodorescu
christodorescu@google.com
Google
Mountain View, CA, USA

Abstract

The major mobile platforms, Android and iOS, have introduced changes that restrict user tracking to improve user privacy, yet apps continue to covertly track users via device fingerprinting. We study the opportunity to improve this dynamic with a case study on mobile fingerprinting that evaluates developers' perceptions of how well platforms protect user privacy and how developers perceive platform privacy interventions. Specifically, we study developers' willingness to make changes to protect users from fingerprinting and how developers consider trade-offs between user privacy and developer effort. We do this via a survey of 246 Android developers, presented with a hypothetical Android change that protects users from fingerprinting at the cost of additional developer effort.

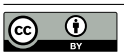
We find developers overwhelmingly (89%) support this change, even when they anticipate significant effort, yet prefer the change be optional versus required. Surprisingly, developers who use fingerprinting are six times more likely to support the change, despite being most impacted by it. We also find developers are most concerned about compliance and enforcement. In addition, our results show that while most rank iOS above Android for protecting user privacy, this distinction significantly reduces among developers very familiar with fingerprinting. Thus there is an important opportunity for platforms and developers to collaboratively build privacy protections, and we present actionable ways platforms can facilitate this.

CCS Concepts

• **Security and privacy** → **Privacy protections**; **Usability in security and privacy**; **Social aspects of security and privacy**; • **Software and its engineering** → *Software design tradeoffs*.

Keywords

privacy, developers, usable privacy, device fingerprinting



CHI '26, Barcelona, Spain

ACM ISBN 979-8-4007-2278-3/2026/04
https://doi.org/10.1145/3772318.3793707

ACM Reference Format:
Alex Berke, Güliz Seray Tuncay, Michael Specter, and Mihai Christodorescu. 2026. Uncovering Relationships between Android Developers, User Privacy, and Developer Willingness to Reduce Fingerprinting Risks. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3772318.3793707

1 Introduction

The major mobile platforms, Apple and Google, have made changes to improve privacy by allowing users to opt out of tracking via advertising IDs [5, 21]. However, mobile apps continue to covertly track users via "device fingerprinting" [15, 30, 49]. Fingerprinting is achieved by collecting device-specific attributes via APIs, and using their combination to identify and track users' devices. Unlike tracking via advertising IDs, fingerprinting can occur without user notice and without an opt-out mechanism, presenting privacy risks that circumvent the control of both users and platforms. Apple's policies disallow tracking users via fingerprinting [5] and both Apple and Google have introduced changes to make this misuse of platform APIs more difficult [4, 19]. While these platform changes are important improvements, the fact that developers continue to circumvent them highlights the critical role developers play in improving or degrading user privacy. Yet the dynamics that guide whether developers adopt or evade policies designed to protect user privacy remain underexplored, which we address through this work.

More specifically, we conduct a fingerprinting study with Android developers. We build on recent works that leverage the open source nature of the Android ecosystem to demonstrate the pervasive privacy risk of fingerprinting [15, 49] (the closed nature of the iOS ecosystem has made large scale iOS evaluations less available [30]). Through this study we evaluate how developers' relationships to fingerprinting impact their perceptions of how well platforms (Android and iOS) protect user privacy, how developers perceive platform changes that protect users from fingerprinting, and how developers consider trade-offs between user privacy and developer effort. We do this by surveying 246 knowledgeable Android developers about a hypothetical platform change, termed "API Usage purposes", described as a way to improve user privacy by

protecting users from fingerprinting. The change comes at the potential cost of additional developer effort — developers must declare their reasons for using APIs that could be abused for fingerprinting in a manifest file [18] — allowing us to study an effort-privacy tradeoff. This change is similar to Apple's "Required Reason API", but no similar mechanism is present in Android at the time of this study. Thus this presents us with the opportunity to study developers' responses to a viable platform intervention by surveying Android developers about this change to the Android ecosystem.

We use our survey to study the following research questions:

RQ1 How do developers trade off developer effort versus user privacy?

RQ2 When introducing privacy-enhancing changes, do developers prefer platforms to mandate requirements or incentivize optional adoption?

RQ3 What are developers' main concerns when platforms introduce privacy changes that require developers' participation?

RQ4 How do developers comparatively perceive Android versus iOS's protection of user privacy, and how does familiarity with fingerprinting impact this perception?

By answering these questions (Section 4) we contribute to our understanding of developers' willingness to adopt platform privacy enhancements. Furthermore, we quantify the effort-privacy tradeoff for developers, in contrast to previous work that only observed this tradeoff qualitatively [33].

We find that the overwhelming majority of developers (89%) supported the intervention, including those who said it would require significant effort. More developers supported an optional implementation model, where developers are incentivized to implement the change, versus a required model. Yet still more developers supported the required model (41.5%) versus no change at all (10.6%).

Further analysis of this support yields both expected and unexpected results. As expected, developers who perceived a higher level of effort to implement the change were less supportive, and developers who perceived a more positive impact on user privacy were more supportive, demonstrating an effort-privacy trade-off. Unexpectedly, developers who use fingerprinting, and would be most impacted by the change, were significantly more supportive of the change. This suggests that the developers most needed to expend effort to reduce user tracking may be (unexpectedly) willing collaborators.

In addition, our analysis of open-ended comments reveals consistent concerns among developers. These include user experience, developer compliance, and platform enforcement, which were topics the survey did not mention.

Finally, our results show that most developers ranked iOS above Android for protecting user privacy. Yet this was less often the case for developers very familiar with fingerprinting. This further indicates that fingerprinting can undermine platform privacy protections and developers' perceptions of it, which we further discuss in Section 5.

2 Related Work

To the best of our knowledge, we are the first to survey how developers' relationships to fingerprinting impacts their approach to user privacy.

2.1 Fingerprinting and User Protections

A large body of research studies how websites, mobile applications, and software development kits (SDKs) fingerprint users [12, 32, 49, 58], and the resulting privacy risks [8]. For example, researchers have shown that fingerprinting is pervasive across the web and mobile app platforms, estimating that more than 25% of the top 10K websites use fingerprinting [27] and more than 19% of the top 30K Android apps do [15]. Other researchers have shown how fingerprinting risks vary across demographic groups, finding risks are higher for lower-income and older US user groups due to the types of devices they use [8].

Furthermore, fingerprinting often circumvents user choice to opt out of tracking [44] by taking advantage of APIs which exist to improve software functionality, violating users' privacy defined by Contextual Integrity [42], which considers the appropriate flow of data under users' expectations. For example, researchers have shown how websites often bypass GDPR protections and use fingerprinting even when users decline cookie consent banners [44]. They have also shown how fingerprinting scripts can be used to restore tracking cookies that users deleted and that this strategy is commonly used across the web, again bypassing users' control [16].

There are also a number of studies that focus on automatic detection and prevention of fingerprinting behavior [13, 27, 43], often treating the developers who use fingerprinting as adversaries. In contrast, we study an opportunity to shift these developers towards the adoption of platform protections.

In mobile apps, many APIs useful for fingerprinting are protected by permissions, where users must grant the app access [38, 56]. App developers declare such permissions in a configuration file, which is the `AndroidManifest.xml` file for Android [18]. Previous surveys have studied how both developers and users engage with permissions [14, 48, 50, 55]. Developers often request excessive permissions due to misunderstanding their scope or the needs of third-party libraries [50]. Similarly, users tend to grant permissions without fully understanding them [14, 48]. Prior work has shown these issues can be reduced for Android by changes within the Google Play Console that "nudge" developers to use fewer permissions [45]. We also study an opportunity for a platform to impact developers' use of permissions, via adding friction rather than nudging.

2.2 Developer Privacy Perceptions

Prior empirical software engineering research has also surveyed developers about their relationships to user privacy [53]. These studies often conclude that developers care about user privacy, but their practices may contradict this sentiment, either because they are unaware of how third-party ads and analytics tooling collect user data [7], or because they see themselves as unable or not responsible to address such privacy risks [39]. When addressing how privacy can be improved, many studies have focused on the role of company culture and communication within teams and lack of awareness of privacy practices [7, 26, 28, 52], advocating for greater adoption of privacy-by-design strategies [24, 46]. When more specifically surveying app developers about user privacy and app permissions, developers have surfaced concerns about how the use of unnecessary permissions can break user trust [50]. Our

survey builds on these results by also focusing on app permission systems and similarly finding contradictions between developers' desire to improve user privacy and their apps' actual practices (i.e. their use of fingerprinting). We further contribute to this literature by measuring developers' preferences for a platform privacy enhancement to reduce fingerprinting, and how their perceptions of privacy benefits impact their preferences.

Other research has studied developer privacy perceptions by analyzing their discussions on online forums [23, 33, 54]. For example, a forum analysis suggested that Android and iOS developers had different interpretations of privacy [23]. Additionally, a case study of the Reddit Android developer forum, `/r/androiddev`, suggested that developers often find new privacy-enhancing restrictions by the platform cause considerable cost yet fail to generate any compelling benefit for developers [33]. The authors suggested Android complement restriction-based approaches with optional approaches that provide nudges [25], via rewards, for developers to make privacy-enhancing changes. In this work we address these qualitative findings and suggestions with quantitative analyses. We quantitatively test how Android developers consider trade-offs between their estimated effort to implement new privacy-enhancing platform changes (i.e., cost) versus impact on user privacy (benefit). Furthermore, we directly query their preference for optional, reward-based changes, versus required change.

3 Materials and Methods

To answer our research questions defined in Section 1, we developed a survey (Section 3.1), recruited Android developer participants (Section 3.3), and analyzed the survey results using a mixed-methods approach (Section 3.4).

3.1 Android Developer Survey

3.1.1 Survey Overview. The survey introduced a hypothetical change to the Android platform, called "API Usage Purposes," designed to protect users from unwanted fingerprinting and improve user privacy. It also queried developers' familiarity with fingerprinting and whether they use fingerprinting in their apps or SDKs.

We limited our survey to an Android change with Android developers because Apple already has the similar "Required Reason API" and part of our goal is to study developers' responses to a new intervention.

After describing API Usage Purposes, our survey solicited concerns, and asked participants how the change would impact user privacy and developer effort. It also asked participants whether they thought Android should make this change as either a requirement, optional, or not at all. We used the responses to study how Android developers trade off developer effort versus user privacy (RQ1), whether developers prefer platforms introduce optional versus required changes to improve user privacy (RQ2), and to identify developers' main concerns when platforms introduce privacy-enhancing changes that require developers' participation (RQ3).

Our survey also gauged participants' sentiments on how well Android and iOS protect user privacy. We used these responses to study RQ4.

3.1.2 Survey Instrument. The survey was administered via Qualtrics. Details on the survey, and all question text, are provided in the Appendix (A).

The survey first asked for participants' informed consent (Q0) and confirmed participants were Android software developers working on either an Android app or software development kit (SDK) (Q1). The survey then included a screening question to assess participants' knowledge of the `AndroidManifest.xml` file and permissions (Q2). Participants who answered incorrectly or "I don't know" were filtered out of the sample used in analysis. The next set of questions (Q3-11) asked participants about how their app or SDK was categorized with Google Play and demographic questions, including age, gender, country, years of experience as a professional developer and developer team size, which were used in related app developer studies [7, 29, 34, 35, 39, 51]. An attention check was also included to improve sample quality.

Before mentioning fingerprinting or "API Usage Purposes", the survey asked how much participants agreed with the following statements: "Android protects user privacy" and "Apple protects user privacy" on a 1-10 scale (Q12).

The survey then described device fingerprinting and asked if participants were already familiar with fingerprinting (Q13). On a following page the survey then asked whether their app or SDK fingerprints users (Q14). To reduce potential response bias, this question reminded participants that we would keep their answers confidential and not attempt to reconnect their responses with their app or SDK. The survey then presented "API Usage Purposes" as a hypothetical change to Android, designed to protect users from unwanted fingerprinting, to improve user privacy. It explained that with this change, Android developers would need to declare purposes for specific APIs that can be used for fingerprinting in their `AndroidManifest.xml` file (Figure 1).

Next, the survey asked participants how much they agreed with the statement that "Android protects user privacy" (with the same 1–10 scale as before), given the assumption that Android required API Usage Purposes (Q16). This repeated question (Q12) was used to measure a potential change in response before versus after API Usage Purposes.

The survey then solicited participants' concerns for API Usage Purposes via open ended comments (Q17a) and asked participants how the change would impact developer effort and user privacy (Q18a-b), with response options on a 5-point Likert scale.

Finally, participants were asked to consider two ways Android could implement the change: an optional model, where apps can receive a user-facing privacy badge and higher rank in the Google Play store, and a required model, where API calls fail when API Usage Purposes are not provided for the impacted APIs. Participants were then asked whether Android should implement the change with either the optional or required model, or not at all (Q19).

3.2 Ethical considerations

The authors' institution did not require an IRB or ethics approval process for this study. However, we took the following steps to respect participants' privacy. First, we made sure participants completed our informed consent form before allowing them to proceed to the survey, which informed them that their data may be used

API Usage Purposes are designed to protect users from unwanted fingerprinting, to improve user privacy.

Android developers would add "purposes" to `<uses-permission>` xml elements in their AndroidManifest.xml file for specific APIs that can be used for fingerprinting.

Here's an example of how *API Usage Purposes* would work.

Suppose you are the developer for an app/SDK that uses the APIs <API_1> and <API_2> from the previous table.

Assume there will be new permissions for each of these APIs.

Your AndroidManifest.xml file would include the following:

```
<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.my.example.app>

  <uses-permission android:name="android.permission.PERMISSION_FOR_API_1">
      <purpose android:name="NotForFingerprinting"/>
  </uses-permission>

  <uses-permission android:name="android.permission.PERMISSION_FOR_API_2">
      <purpose android:name="NotForFingerprinting"/>
  </uses-permission>
  ...
</manifest>
```

Assume tools within Android Studio will help developers fill in the syntax for each API with automatic suggestions.

Figure 1: Screenshot from the developer survey (directly before Q16), which explains how developers would implement the hypothetical change with an example `AndroidManifest.xml` code snippet. <API_1> and <API_2> serve as example APIs that would be impacted by the change.

in a research publication and would be anonymized. We then only retained survey responses from participants who completed the entire study. Furthermore, we assigned random participant IDs and deleted any PII and other data that could be used to re-identify participants, including the app or SDK they worked on.

All participants who completed the survey were compensated, whether or not they passed the attention check or screening.

3.3 Participants

We recruited English-speaking Android developer participants from two panels that are managed by a third-party vendor contracted by our company. The first is a panel of mobile app and web developers who were recruited via banner ads placed on https://web.dev and https://developer.chrome.com, which host documentation for software developers. These developers were offered $2.50 USD for completing the survey. The second is a panel of Android developers. They had opted-in to share their email addresses for research and marketing outreach when creating their Google Play accounts, and were later recruited for the panel via their Play account email addresses. These developers were offered $5 for completing the survey (with the higher amount going to developers with already verified identities). Both panel recruitment processes included a pre-screening to ensure participants were professional developers in English speaking countries. We further used our survey screening to help ensure pre-screened participants were Android developers working on an app or SDK. We note the use of targeted web ads and outreach via Google Play accounts has similarly been used to recruit developer participants for prior research studies [29, 51]. All participants completed our survey in June 2025 and were paid whether or not they were screened out. On average the survey took participants 7 minutes to complete.

A total of 498 participants began the survey after providing informed consent. Our screening then excluded 124 who were not Android developers working on an Android app or SDK, 117 who did not pass the knowledge assessment for the `AndroidManifest.xml` file and permissions, and 11 who failed the attention check. The resulting data sample we analyzed includes the $N = 246$ participants who passed all screening.

Details about the sample are in Appendix A and our anonymized survey dataset can be accessed upon request for research purposes. Of the 246 participants, 214 (87%) were male, 23 (9.3%) female, and the remainder answered "Other" or "Prefer not to answer." We note this gender imbalance is consistent with developer samples from

prior work, where women represent 1–26% [7, 29, 39, 50]. When grouped by age, 58 (23.6%) were 18–34 years old, 158 (64.2%) 35–54 years old, and 27 (11%) 55 years or older. The majority of participants had at least three years of professional Android developer experience and worked on teams of fewer than five developers. When grouped by country, the largest groups (35.8%) were from the US, then the UK (17.1%), India (13.4%), Canada (6.9%), and Germany (6.5%).

Figure 2 summarizes participants' relationships to fingerprinting, with comparisons across app versus SDK developers (see also Appendix Tables 7-8). 228 (92.7%) of the participants said they primarily work on an app while 18 (7.3%) said they primarily work on an SDK (Q3). When asked whether they were already familiar with device fingerprinting (Q13), 91 (37%) answered "Very familiar", 123 (50%) answered "Somewhat familiar", 26 (10.6%) answered they were not previously familiar, but now understood it, and 6 (2.4%) answered that even after the explanation, they did not understand it. When asked whether their app/SDK fingerprints users (Q14), the majority, 49 (60.6%) answered "No", while 44 (17.9%) answered "Yes", 44 (17.9%) answered "No directly, but a dependency does", and 11 (4.5%) answered "I'm not sure". Figure 2 shows how the majority of those who said they are "very familiar" with fingerprinting likely did so because their app/SDK uses fingerprinting, while the majority who said they do not use fingerprinting answered that they were only "somewhat familiar" or "previously unfamiliar" with fingerprinting. It also shows that the percentage of SDK developers who use fingerprinting directly (38.9%) is more than twice that of app developers (15.4%), and more app developers said their app does fingerprinting via a dependency (i.e. SDK) versus directly (18% versus 15.4%). This is consistent with prior research that used program analysis to measure fingerprinting in Android apps and highlighted SDKs as the primary source of fingerprinting [49]. (For further context on developers' use of fingerprinting and their app/SDK category, see Appendix Tables 5-6). We used these responses in the following analyses to further study participants' relationships to fingerprinting.

3.4 Study Analysis

3.4.1 Quantitative Analysis. We hypothesized that developers who use fingerprinting in their apps and SDKs, and who would therefore be most impacted by a change like "API Usage Purposes", would be least supportive of this change.

We used logistic regression to test this hypothesis and study how developers trade off effort and user privacy (RQ1). We mapped responses to the survey question about the level of effort required to implement API Usage Purposes (Q18a), which was asked with a 5-level Likert scale, to a 1 (very little effort) to 5 (very significant effort) scale. We similarly mapped responses to the question about impact on user privacy (Q18b) to a 1 (very negative impact) to 5 (very positive impact) scale. We also created a binary variable indicating whether the app or SDK they work on fingerprints users (Q14), mapping responses for either directly or via a dependency to 1, 0 otherwise. We used these as independent variables in the regression model. The dependent variable was whether or not they answered "Yes" when asked if Android should implement API Usage Purposes (Q19). This includes yes to either the required or optional model. For robustness, we repeated this analysis after excluding responses that answered "Yes" to the optional model, in order to restrict the analysis to testing support for a required change.

To analyze how developers comparatively perceived Android and iOS's protection of user privacy, we mapped their levels of agreement to the statements "[Apple/Android] protects user privacy" to whether they ranked Android above iOS, iOS above Android, or equivalently (Q12). We used logistic regression to study how their familiarity with fingerprinting (Q13) impacted this outcome by using whether they said they were "very familiar with fingerprinting" (n=91) as a binary, independent variable, and whether they ranked iOS above Android as the dependent variable.

We then measured the change in their response to "Android protects user privacy" at the end of the survey, after they were asked to assume API Usage Purposes were implemented (comparing Q16 to Q12). We created a binary variable indicating whether their level of agreement with this statement increased (1) or not (0), and used this as the dependent variable in another logistic regression model which included whether they were "very familiar with fingerprinting" as an independent variable.

As an additional robustness check, we repeated all of the above regression analyses with variables to control for demographics (age and gender) as well as team size, years of professional developer experience, and whether the developer worked on an app versus SDK.

All analyses were conducted using Python and the pandas [11] and statsmodels [47] libraries.

3.4.2 Qualitative Analysis of Comments. To study RQ3, the survey asked participants to optionally provide their concerns for API Usage Purposes (Q17a), resulting in 184 written responses. To evaluate these responses the four authors independently, inductively open-coded all responses using thematic analysis [40], identifying themes, with some written responses having one theme and others having multiple themes. They then discussed and merged their codebooks and aligned on six themes, provided in Appendix Table 19. Two authors then acted as coders and independently went through the responses again to label each with the top most relevant theme, and then measured inter-rater reliability (where the calculation excludes blank responses). The resulting Cohen's kappa coefficient is 0.74 which is considered substantial agreement [31]. The two authors resolved the remaining disagreements through discussions to produce the final labels presented in the findings (Section 4.2).

4 Findings

Here we describe our findings, including developers' support for changes (Section 4.1), their concerns (Section 4.2), and perceptions of privacy in iOS versus Android (Section 4.3).

4.1 Support for Changes to Improve User Privacy (RQ1 and RQ2)

Developers in our sample overwhelmingly supported the hypothetical API Usage Purposes. When asked whether Android should implement the change, 102 (41.5%) said "Yes, with the required model," 118 (48.0%) said "Yes, with the optional model," and 26

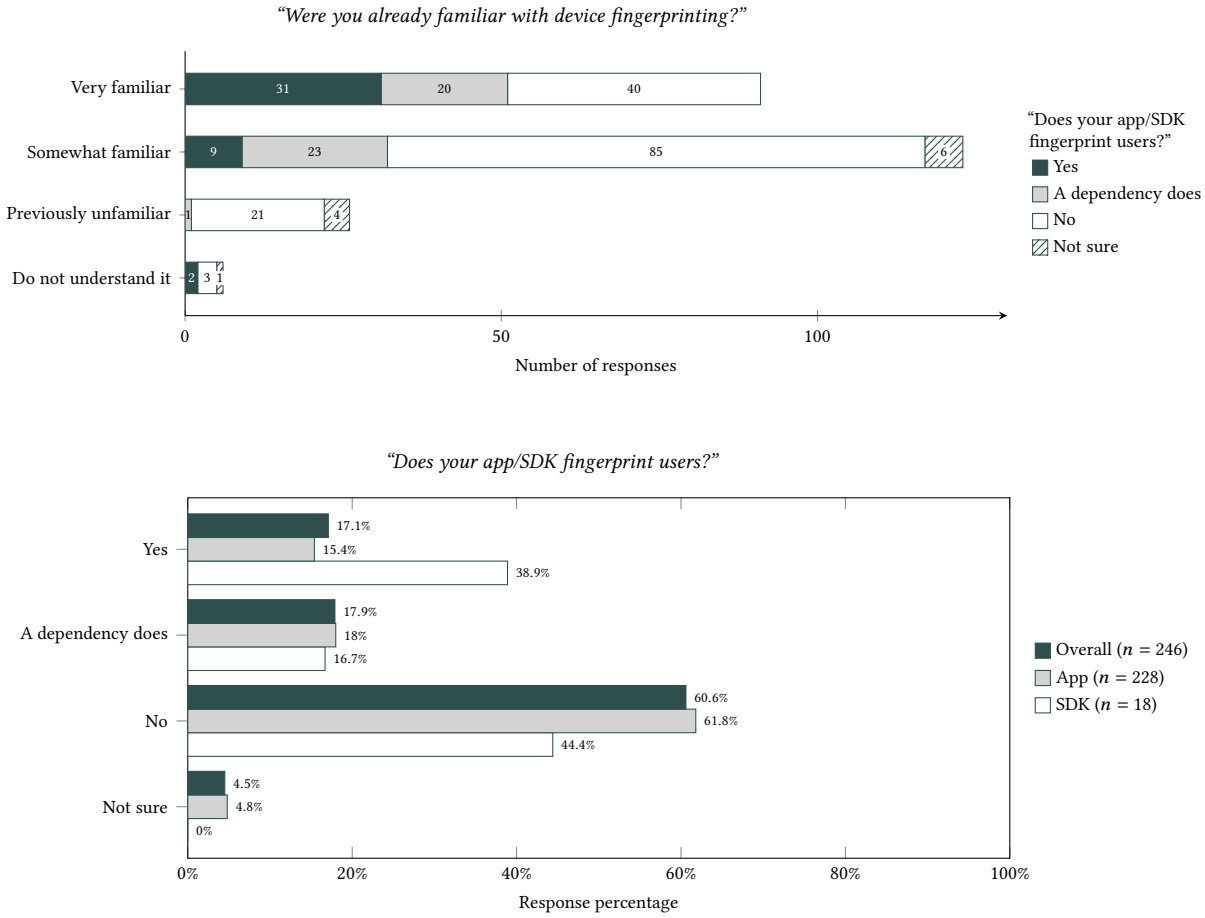


Figure 2: Participants' relationships with fingerprinting.

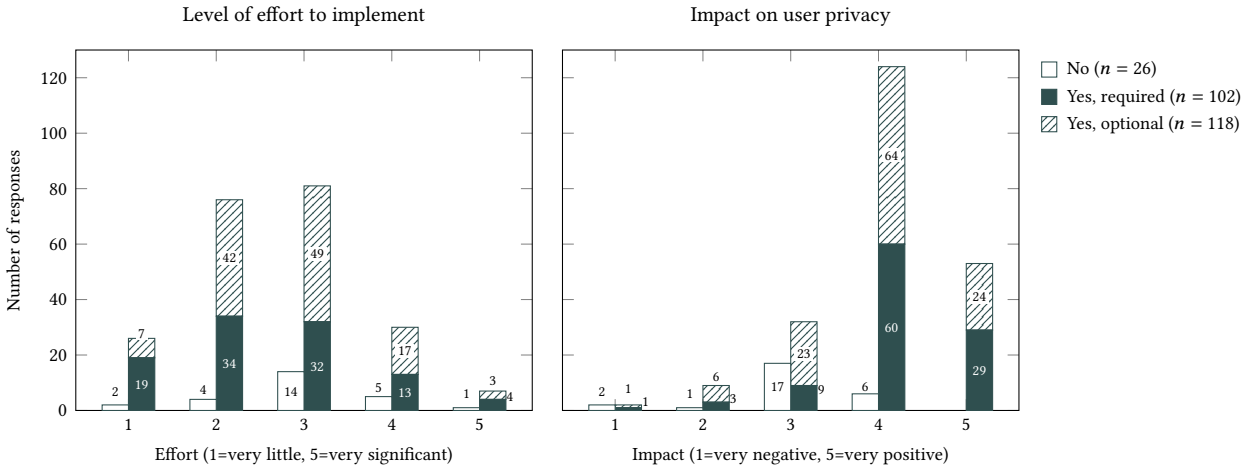


Figure 3: Responses to whether Android should implement the change by levels of perceived developer effort (left) and impact on user privacy (right).

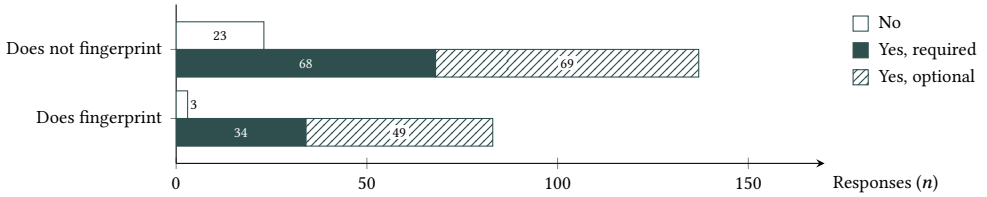


Figure 4: Responses to whether Android should implement the change by whether the developer said their app/SDK fingerprints users.

(10.6%) said "No". These results show more support for the optional versus required model, yet even so, nearly 4 times as many participants supported a required change versus no change at all. Furthermore, even developers who said the change would require moderate to significant effort to implement were more likely to support the change.

Most developers we surveyed anticipated the change would require very little to moderate effort, and most said the change would have a positive or very positive impact on user privacy. Results to these survey questions are shown in Figure 3 and Appendix Tables 14 and 15. Our regression analysis shows developers' perceptions of the impact of the change on developer effort and user privacy significantly affects their support for the change. Higher levels of perceived effort negatively impacted support ($OR = 0.46$; $p < 0.01$) and more positive perceptions of the impact on user privacy positively impacted support ($OR = 3.651$; $p < 0.001$).

Our regression analysis also tested whether developers whose apps or SDKs use fingerprinting were more or less likely to support API Usage Purposes. Contrary to our hypothesis, developers who reported using fingerprinting were more than 6 times as likely to support the change ($OR = 6.480$; $p < 0.05$, Figure 4). Regression results are shown in Table 1. As a robustness check, we repeated the analysis after filtering out participants who answered with support for the optional model, in order to directly compare the responses for "Yes, with the required model" to "No, not at all." The regression results are directionally consistent, yet the effect sizes are even larger; developers whose apps/SDKs use fingerprinting were even more likely ($OR = 8.288$; $p < 0.05$) to support the change (see Appendix Table 17). We also performed a robustness check with variables indicating whether participants work on an app versus SDK and their demographics. The results are consistent and we did not find working on an app versus SDK had a significant impact (Appendix Table 18).

4.2 Developer Concerns (RQ3)

Figure 5 summarizes the top themes that emerged from our inductive analysis of the 184 open-ended responses (Section 3.4.2). 71 of these responses came from developers who use fingerprinting, either directly or via a dependency, which we indicate with an asterisk (*). 115 (46.7%) participants did not list a concern about API Usage Purposes. This includes 62 blank responses and 53 responses that explicitly stated no concern or positive support; some listed concerns about other parts of the Google/Android ecosystem. Examples include *"Seems fine to me" (P180)*, *"This is a good idea" (P20)*, and *"No concerns. It's a good idea. It's rather Google that doesn't respect privacy" (P33)*.

Table 1: Logistic regression model where the dependent variable is whether the developer said "Yes" to whether they think Android should implement the change. Note: $*\ p < 0.05$; $**\ p < 0.01$; $***\ p < 0.001$.

Variable	Coef.	OR	95% CI	p-value
Intercept	−0.716	0.489	[0.061, 3.936]	0.501
Effort level	−0.777**	0.460	[0.270, 0.782]	0.004
Privacy impact	1.295***	3.651	[2.153, 6.189]	< 0.001
Does fingerprint	1.869**	6.480	[1.656, 25.354]	0.007
N	246			
Pseudo R-squared	0.2514			

4.2.1 Compliance and enforcement. The most common theme regarded compliance and enforcement, however this theme was more common among developers who do not use fingerprinting (N=25 versus N*=9). Participants worried that developers would lie about whether they used APIs to fingerprint users, or would find other ways to evade the policy. For example, *"Seems annoying and anyone can just lie, no?" (P4)*.

In particular, many participants voiced concerns that the Google Play Store would not properly detect or validate the use, or lack of use, of API Usage Purposes. For example, *"Will Google Play Store validates it's used correctly?" (P51)*.

They were also concerned about lack of enforcement, or that enforcement would be arbitrary. For example, *"Google Play's enforcement of these XML tags will be arbitrary. I do not trust that having developer-declared usage will solve the fingerprinting problem" (P16)* and *"How it will be enforced. Every change like this ends up with a mess on the review process on the Play Store Console" (P72)*. Our survey did not mention compliance or enforcement, so the fact that so many participants independently voiced this concern should bring attention to it.

4.2.2 Developer and engineering challenges. 26 participants highlighted potential developer and engineering challenges. These concerns were solicited before our survey asked participants to evaluate the level of effort required to implement API Usage Purposes and were more common for developers who use fingerprinting (N*=14 versus N=12), which might be expected given they would be more impacted by the changes. These concerns included how the change might limit the legitimate use of APIs, such as *"More restriction*

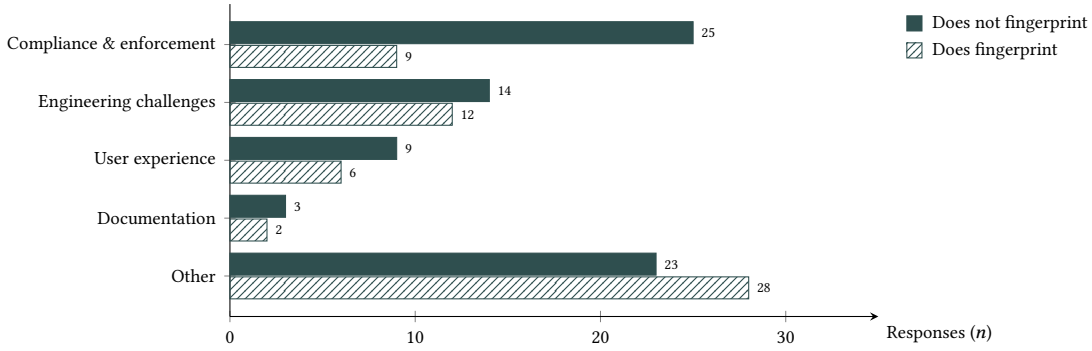


Figure 5: Top themes for developer concerns provided via open-ended responses

could make it harder to use when there is a legitimate need for that information" (P8).

Concerns also covered backwards compatibility, the increasing difficulty of building or testing an app, the potential for changes to break app functionality, and overall developer workload. Examples include *"API usage makes it easier to build the App. Any restrictions will increase the QA efforts" (P9*)*, *"The potential issues with API and SDK calls not working properly" (P101*)*, and *"The additional work, risk of getting it wrong" (P42).*

4.2.3 User experience. Our survey did not mention user experience, leaving it undetermined whether users would be made aware of API Usage Purposes. Yet 15 participants still brought up concerns regarding the users' perceptions or experiences. For example, *"Could lead to issues where the user is concerned about something . . . that was never an issue for them to be concerned about the first place" (P151)*, *"Adding extra layers of information could lead to overwhelming the user" (P113)* and *". . . vague or misleading descriptions, user fatigue from too many prompts. . . " (P165*).*

4.2.4 Documentation. 5 participants also brought up concerns regarding clarity and documentation of how API Usage Purposes should be used. Most of these participants first listed support for the change. For example, *"I do like very much that the permission can be declared for finger printing or not. Besides that, good documentation of how to implement this changes is paramount" (P123*)* and *"I don't have any. Just make sure it's well documented" (P112).*

4.2.5 Other. 51 participants voiced concerns that could not be grouped into consistent themes. Examples include *"Unclear why it's needed" (P97)* or *"I think it's a nice improvement. I use AdMob, though, so I wonder how it would affect my advertising revenue" (P88*).*

4.3 Perceptions of Privacy in iOS versus Android (RQ4)

We evaluated participants' levels of agreement to the statements "[Apple/Android] protects user privacy." The median level of agreement was the same (*median* = 8) for Apple and Android (Figure 6). However, the mean was higher for Apple (*mean* = 7.9; *SD* = 2.1) versus Android (*mean* = 7.3; *SD* = 2.2).

When making participant-level comparisons, we found that more participants ranked Apple above Android versus the other way around (see Figure 6 and Appendix Table 10) . However, we also found that developers' familiarity with fingerprinting played a significant role in this ranking (Figure 7). Developers who said they were very familiar with fingerprinting were significantly less likely to rank Apple above Android with respect to user privacy ($OR = 0.540$; $p < 0.05$). This finding holds when controlling for developer demographics and whether they work on an app versus SDK, which did not play a significant role. See Appendix B for details on the regression (Table 11) and a comparison in the distributions of participants' agreement levels (Table 10).

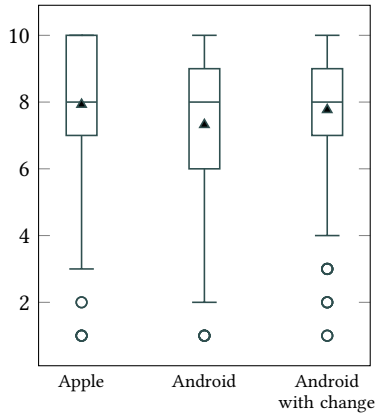


Figure 6: Boxplots showing the distribution of participants' levels of agreement from 1 (strongly disagree) to 10 (strongly agree) for the statements "[Apple/Android] protects user privacy." Lines indicate 25th, 50th (median) and 75th percentiles; triangles indicate means.

When participants were again asked about how "Android protects user privacy" after assuming API Usage Purposes were implemented, the median value did not change and the mean value slightly increased (*mean* = 7.8; *SD* = 2.0). We did not find that developers' familiarity with fingerprinting played a significant role in this increase (see regression results in Appendix Table 13).

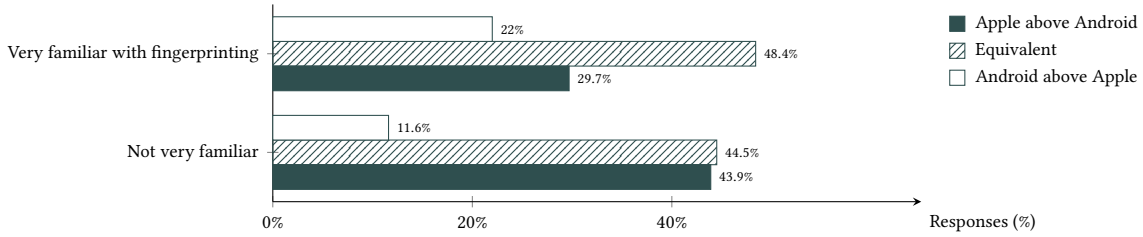


Figure 7: Participant-level comparisons for levels of agreement to whether "[Apple/Android] protects user privacy," by whether or not they are very familiar with fingerprinting.

5 Discussion

Our findings can provide insights for platforms as they make privacy-enhancing changes that affect both developers and end-users. We find developers' relationships to fingerprinting impact their perceptions of how well platforms protect user privacy, and their willingness to participate in improving it.

5.1 Developer Concerns about Enforcement and User Experience

Our survey did not mention compliance and enforcement, or user experience, yet these themes emerged from the open ended comments left by developers.

Compliance and enforcement. Many developers voiced concerns that developers would lie about the true API purposes and that enforcement would be poorly handled by the platform. As platforms make changes impacting user trust, these concerns highlight how they should also consider developer trust.

User experience. Many developers were concerned that the change, if presented to users, would overwhelm them. Prior work has shown [1] that over-prompting users with warnings can cause fatigue, effectively training users to ignore and click through otherwise important guardrails. It is unclear if, when, and how to present users with information like API Usage Purposes and thus it remains a question to address in future work.

5.2 Developers Support Changes to Improve User Privacy Despite Additional Effort

Our work addresses prior research that suggested Android provide incentive-based approaches when making privacy-related changes [33]. Indeed, developers we surveyed preferred an optional, incentive-based change over a requirement. Our work also addresses the issue of cost for developers when Android introduces such changes [33]. We found overwhelming support for "API Usage Purposes," despite additional effort required by developers.

Furthermore, contrary to our hypothesis (Section 3.4.1), **we found developers who use fingerprinting were six times more likely to support the change**, despite potential impact to their business model. A possible explanation is that, while many developers care about user privacy, developers who use fingerprinting are much more aware of the negative impact this practice has on user privacy. Yet they may also need to engage in fingerprinting to maintain a competitive edge, since so many of their competitors use this highly effective user tracking method. While our survey did not address why developers use fingerprinting, previous work has demonstrated how fingerprinting provides data for precise ad targeting [36], which is economically valuable [57]. This scenario presents a collective action problem that a platform change like API Usage Purposes could help resolve by impacting all apps at once. Regardless of why developers who use fingerprinting support a change to reduce fingerprinting, this surprising result is heartening – developers could support the adoption of such privacy enhancements, if the barriers we found (e.g. documentation, compliance, and enforcement) were overcome.

5.3 Different Perspectives on Platform Privacy

On average, developers we surveyed ranked Apple slightly higher than Android for protecting user privacy, yet the rankings were very similar (the medians identical). This similarity is more reflective of a large-scale empirical analysis of Android and iOS apps that found widespread user tracking and privacy violations in both ecosystems [30].

The developers in our survey who indicated strong familiarity with fingerprinting were significantly less likely to rank Apple above Android for protecting user privacy. While further surveys are needed to explain this discrepancy, we provide some speculation. Apple has marketed itself as a more private platform [10, 37] and Apple made a highly publicized [41] change in iOS 14, called "App Tracking Transparency," which requires apps to directly request the user's permission before tracking them with advertising IDs [2, 3]. While this may improve Apple's privacy sentiment for many users and developers less familiar with fingerprinting, those who understand how fingerprinting bypasses user tracking controls may feel differently. Regardless of the reason for this discrepancy, our findings suggest that Android could improve its comparative privacy sentiment by improving education and awareness around fingerprinting.

5.4 Opportunity and Recommendations for Platform–Developer Collaboration on User Privacy

Our results are consistent with prior work indicating app developers care about user privacy, even when their actions may be contradictory [7, 39]. They also echo prior work that calls for increasing collaboration between developers and API designers rather than treating developers as enemies [9]. Altogether, these results present

an opportunity for platform-developer collaboration to improve user privacy.

In particular, we recommend platforms make changes to more directly address how fingerprinting undermines user privacy, which our survey showed most Android developers are ready to support. These changes can be motivated by developers' and platforms' shared incentive of increasing user trust and perceived privacy – developers care about user trust [50] and privacy [7, 39], and the major platforms consistently highlight the ways they improve user privacy [6, 22]. These changes can be further supported by better educating users and developers about fingerprinting, as noted in Section 5.3. In terms of how to make the changes to address fingerprinting risks, API Usage Purposes can serve as an example, as well as provide an example for platform-developer collaboration on improving user privacy more broadly.

With any such changes we recommend platforms collaboratively incorporate developer feedback, such as the concerns solicited via this survey. For example, platforms should help alleviate developer burden and engineering challenges by providing tools (e.g., through their IDEs) that simplify implementation. Platforms should also support developers with clear documentation that includes how changes will impact user experiences. They can also support developers with well defined policies and mechanisms to detect and enforce violations that developers can trust.

5.5 Limitations and Future work

The scope of our survey is limited to Android developers and questions about fingerprinting protections are limited to a specific proposal. Future work should address generalizability. This includes testing whether our findings on developer perceptions of user privacy extend to iOS developers. In particular, our survey did not ask if participants were familiar with iOS development or iOS's Required Reason API [4], which could have informed their responses about the platforms' user privacy protections or "API Usage Purposes". To address these limitations, future work can survey both Android and iOS developers and also ask iOS developers about their actual experiences using the Required Reason API. Another limitation is that our survey relied on self-reported data and described a hypothetical change at a high level with pseudocode. It could be valuable to validate participants' reported relationships to fingerprinting by drawing information from their apps/SDKs and validate our findings in real-world development settings. (We did not connect participants to their apps/SDKs to protect their privacy and minimize their self-censorship.)

Potential biases in survey responses also present limitations. For example, social desirability bias could lead participants who use fingerprinting to say they do not. However, our result that developers who use fingerprinting are more likely to support a change to reduce fingerprinting holds despite this potential bias, which we expect would weaken this result. Even so, developers we surveyed opted in to participate, and this self-selection may present bias.

Finally, our survey raises important questions that remain unanswered. If developers are ready to adopt changes to protect users from fingerprinting, why do many of them use fingerprinting? It is possible that developers have little control over the fingerprinting done by SDKs that their apps use, or fear losing the competitive advantage that fingerprinting provides. If developers do not wish to lose the benefits of fingerprinting and user tracking (e.g. increased advertising revenue [36, 57]), the platforms may need to guide a collective change in developer behavior. While "API Usage Purposes" may present one way, understanding the optimal mechanisms to effect such changes is the topic of further research.

6 Conclusion

Our study evaluated developers' willingness to adopt a privacy-enhancing platform change via a survey with 246 knowledgeable Android developers. The survey introduced a hypothetical change to improve user privacy by reducing fingerprinting risks, with a potential cost to developers.

Our contributions include measuring an effort-privacy trade-off and finding developers are overwhelmingly supportive of such a privacy-enhancing change, even when they anticipate significant effort. Furthermore, we found developers who use fingerprinting, and would be most impacted by the change, were surprisingly most likely to support it. Our analysis of developers' comments also highlights their concerns around platform policy enforcement, providing guidance for platforms as they consider such changes. Overall, this study reveals an important opportunity for platforms to improve user privacy with a collaborative model that addresses developer concerns and guides collective action.

Acknowledgments

We thank the Android developers who participated in our survey for their contributions. We also thank Mark Cwalinski and Matt Warner for helping motivate and develop our research questions, Richard Smith for helping recruit developers, and the anonymous reviewers for providing helpful feedback to clarify the work. We are thankful to Hamzeh Zawawy, René Mayrhofer, and Dave Kleidermacher for their feedback and support throughout this project.

References

[1] Devdatta Akhawe and Adrienne Porter Felt. 2013. Alice in Warningland: A large-scale field study of browser security warning effectiveness. In *22nd USENIX Security Symposium (USENIX Security 13)*. Washington, D.C., 257–272.

[2] Apple. 2021. Data Privacy Day at Apple: Improving transparency and empowering users. https://www.apple.com/newsroom/2021/01/data-privacy-day-at-apple-improving-transparency-and-empowering-users/. Accessed: 2025-08-18.

[3] Apple. 2025. App tracking transparency. https://developer.apple.com/documentation/apptrackingtransparency. Accessed: 2025-08-18.

[4] Apple. 2025. Describing use of required reason API. https://developer.apple.com/documentation/bundleresources/describing-use-of-required-reason-api. Accessed: 2025-09-18.

[5] Apple. 2025. User privacy and data use. https://developer.apple.com/app-store/user-privacy-and-data-use/&sa=D&source=docs&ust=1755557882327604&usg=AOvVaw2RjXNDNbeMC-VLccg4V9Hj. Accessed: 2025-09-18.

[6] Apple Inc. 2026. Privacy - Apple. https://www.apple.com/privacy/ Accessed: 2026-01-26.

[7] Rebecca Balebako, Abigail Marsh, Jialiu Lin, Jason Hong, and Lorrie Cranor. 2014. The privacy and security behaviors of smartphone app developers. In *Workshop on Usable Security*. 1–10. doi:10.14722/usec.2014.23006

[8] Alex Berke, Enrico Bacis, Badih Ghazi, Pritish Kamath, Ravi Kumar, Robin Lassonde, Pasin Manurangsi, and Umar Syed. 2025. How unique is whose web browser? The role of demographics in browser fingerprinting. In *Proceedings of the 25th Privacy Enhancing Technologies Symposium (PETS 2025)*. PoPETS, Washington, DC, USA, 720—-758.

[9] Partha Das Chowdhury, Joseph Hallett, Nikhil Patnaik, Mohammad Tahaei, and Awais Rashid. 2021. Developers are neither enemies nor users: They are collaborators. In *2021 IEEE Secure Development Conference (SecDev)*. IEEE, Atlanta, GA,

USA, 47–55. doi:10.1109/SecDev51306.2021.00023
[10] Matt Clinch. 2014. Apple's Tim Cook takes a swipe at Google, Facebook. https://www.cnbc.com/2014/09/18/apples-tim-cook-takes-a-swipe-at-google-facebook.html. Accessed: 2025-08-18.
[11] The Pandas development team. 2025. *pandas-dev/pandas: Pandas*. doi:10.5281/zenodo.15831829
[12] Peter Eckersley. 2010. How unique is your web browser?. In *Proceedings of the 10th International Conference on Privacy Enhancing Technologies* (Berlin, Germany) *(PETS'10)*. Springer-Verlag, Berlin, Heidelberg, 1–18.
[13] William Enck, Peter Gilbert, Seungyeop Han, Vasant Tendulkar, Byung-Gon Chun, Landon P. Cox, Jaeyeon Jung, Patrick McDaniel, and Anmol N. Sheth. 2014. TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones. *ACM Trans. Comput. Syst.* 32, 2, Article 5 (June 2014), 29 pages. doi:10.1145/2619091
[14] Adrienne Porter Felt, Elizabeth Ha, Serge Egelman, Ariel Haney, Erika Chin, and David Wagner. 2012. Android permissions: User attention, comprehension, and behavior. In *Proceedings of the Eighth Symposium on Usable Privacy and Security* (Washington, D.C.) *(SOUPS '12)*. Association for Computing Machinery, New York, NY, USA, Article 3, 14 pages. doi:10.1145/2335356.2335360
[15] Christof Ferreira Torres and Hugo Jonker. 2018. Investigating fingerprinters and fingerprinting-alike behaviour of Android applications. In *European Symposium on Research in Computer Security*. Springer, 60–80.
[16] Imane Fouad, Cristiana Santos, Arnaud Legout, and Nataliia Bielova. 2022. My cookie is a phoenix: detection, measurement, and lawfulness of cookie respawning with browser fingerprinting. In *Privacy Enhancing Technologies Symposium*.
[17] Google. 2025. Google Play SDK index. https://play.google.com/sdks. Accessed: 2025-09-18.
[18] Google. 2025. The manifest file. https://developer.android.com/guide/components/fundamentals#Manifest. Accessed: 2025-09-18.
[19] Google. 2025. Overview of the SDK Runtime on Android Privacy Sandbox. https://privacysandbox.google.com/private-advertising/sdk-runtime. Accessed: 2025-09-18.
[20] Google. 2025. Play Console Help. https://support.google.com/googleplay/android-developer/answer/9859673. Accessed: 2025-09-18.
[21] Google. 2025. Usage of Android advertising ID. https://support.google.com/googleplay/android-developer/answer/9857753. Accessed: 2025-09-18.
[22] Google LLC. 2026. Android Privacy - Safety and Security. https://www.android.com/intl/en_us/safety/privacy/ Accessed: 2026-01-26.
[23] Daniel Greene and Katie Shilton. 2018. Platform privacies: Governance, collaboration, and the different meanings of "privacy" in iOS and Android development. *New Media & Society* 20, 4 (2018), 1640–1657. doi:10.1177/1461444817702397
[24] Irit Hadar, Tomer Hasson, Oshrat Ayalon, Eran Toch, Michael Birnhack, Sofia Sherman, and Arod Balissa. 2018. Privacy by designers: Software developers' privacy mindset. *Empirical Software Engineering* 23, 1 (01 Feb 2018), 259–289. doi:10.1007/s10664-017-9517-1
[25] D. Halpern. 2015. *Inside the nudge unit: How small changes can make a big difference*. W.H. Allen, London, UK.
[26] Stefan Albert Horstmann, Samuel Domiks, Marco Gutfleisch, Mindy Tran, Yasemin Acar, Veelasha Moonsamy, and Alena Naiakshina. 2024. "Those things are written by lawyers, and programmers are reading that." Mapping the communication gap between software developers and privacy experts. *Proceedings on Privacy Enhancing Technologies* 2024 (2024), 151–170. Issue 1. doi:10.56553/popets-2024-0010
[27] Umar Iqbal, Steven Englehardt, and Zubair Shafiq. 2021. Fingerprinting the fingerprinters: Learning to detect browser fingerprinting behaviors. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA, 1143–1161. doi:10.1109/SP40001.2021.00017
[28] Leonardo Horn Iwaya, Muhammad Ali Babar, and Awais Rashid. 2023. Privacy engineering in the wild: Understanding the practitioners' mindset, organizational aspects, and current practices. *IEEE Transactions on Software Engineering* 49, 9 (2023), 4324–4348. doi:10.1109/TSE.2023.3290237
[29] Harjot Kaur, Sabrina Klivan, Daniel Votipka, Yasemin Acar, and Sascha Fahl. 2022. Where to recruit for security development studies: Comparing six software developer samples. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 4041–4058. https://www.usenix.org/conference/usenixsecurity22/presentation/kaur
[30] Konrad Kollnig, Anastasia Shuba, Reuben Binns, Max Van Kleek, and Nigel Shadbolt. 2022. Are iPhones really better for privacy? A comparative study of iOS and Android apps. *Proceedings on Privacy Enhancing Technologies* 2022, 2 (2022), 6–24. doi:10.2478/popets-2022-0033
[31] J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics* 33, 1 (1977), 159–174. http://www.jstor.org/stable/2529310
[32] Pierre Laperdrix, Nataliia Bielova, Benoit Baudry, and Gildas Avoine. 2020. Browser fingerprinting: A survey. *ACM Trans. Web* 14, 2, Article 8 (April 2020), 33 pages. doi:10.1145/3386040
[33] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I. Hong. 2021. How developers talk about personal data and what it means for user privacy: A case study of a developer forum on Reddit. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW3, Article 220 (Jan. 2021), 28 pages. doi:10.1145/3432919
[34] Mario Linares-Vasquez, Christopher Vendome, Qi Luo, and Denys Poshyvanyk. 2015. How developers detect and fix performance bottlenecks in Android apps. In *Proceedings of the 2015 IEEE International Conference on Software Maintenance and Evolution (ICSME) (ICSME '15)*. IEEE Computer Society, USA, 352–361. doi:10.1109/ICSM.2015.7332486
[35] Mario Linares-Vásquez, Cárlos Bernal-Cardenas, Kevin Moran, and Denys Poshyvanyk. 2017. How do developers test Android applications?. In *2017 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 613–622. doi:10.1109/ICSME.2017.47
[36] Zengrui Liu, Jimmy Dani, Yinzhi Cao, Shujiang Wu, and Nitesh Saxena. 2025. The First Early Evidence of the Use of Browser Fingerprinting for Online Tracking. In *Proceedings of the ACM on Web Conference 2025* (Sydney, NSW, Australia) *(WWW '25)*. Association for Computing Machinery, New York, NY, USA, 4980–4995. doi:10.1145/3696410.3714548
[37] Kelly D. Martin and Patrick E. Murphy. 2017. The role of data privacy in marketing. *Journal of the Academy of Marketing Science* 45, 2 (01 Mar 2017), 135–155. doi:10.1007/s11747-016-0495-4
[38] René Mayrhofer, Jeffrey Vander Stoep, Chad Brubaker, Dianne Hackborn, Bram Bonné, Güliz Seray Tuncay, Roger Piqueras Jover, and Michael A Specter. 2024. The Android platform security model (2023).
[39] 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)*. USENIX Association, San Francisco, CA, USA, 225–244.
[40] Matthew Miles, Michael Huberman, and Johnny Saldaña. 2013. *Qualitative data analysis: A methods sourcebook*. SAGE Publications, Inc, Thousand Oaks, CA.
[41] Daniel Newman. 2022. Apple, Meta and the $10 billion impact of privacy changes. https://www.forbes.com/sites/danielnewman/2022/02/10/apple-meta-and-the-ten-billion-dollar-impact-of-privacy-changes/. Accessed: 2025-08-18.
[42] Helen Nissenbaum. 2009. *Privacy in context: Technology, policy, and the integrity of social life*. Stanford University Press, USA.
[43] Gerald Palfinger. 2025. AndroPROTECT: Hardening the Android API against fingerprinting. In *Network and System Security*, Houbing Herbert Song, Roberto Di Pietro, Saed Alrabaee, Mohammad Tubishat, Mousa Al-kfairy, and Omar Alfandi (Eds.). Springer Nature Singapore, Singapore, 39–59.
[44] Emmanouil Papadogiannakis, Panagiotis Papadopoulos, Nicolas Kourtellis, and Evangelos P. Markatos. 2021. User tracking in the post-cookie era: How websites bypass GDPR consent to track users. In *Proceedings of the Web Conference 2021* (Ljubljana, Slovenia) *(WWW '21)*. Association for Computing Machinery, New York, NY, USA, 2130–2141. doi:10.1145/3442381.3450056
[45] Sai Teja Peddinti, Igor Bilogrevic, Nina Taft, Martin Pelikan, Úlfar Erlingsson, Pauline Anthonysamy, and Giles Hogben. 2019. Reducing permission requests in mobile apps. In *Proceedings of the Internet Measurement Conference* (Amsterdam, Netherlands) *(IMC '19)*. Association for Computing Machinery, New York, NY, USA, 259–266. doi:10.1145/3355369.3355584
[46] Maxwell Prybylo, Sara Haghighi, Sai Teja Peddinti, and Sepideh Ghanavati. 2024. Evaluating privacy perceptions, experience, and behavior of software development teams. In *Proceedings of the Twentieth USENIX Conference on Usable Privacy and Security* (Philadelphia, PA, USA) *(SOUPS '24)*. USENIX Association, USA, Article 6, 20 pages.
[47] Skipper Seabold and Josef Perktold. 2010. Statsmodels: Econometric and statistical modeling with Python. In *Proceedings of the 9th Python in Science Conference*, Stéfan van der Walt and Jarrod Millman (Eds.). SciPy.org, https://doi.org/10.25080/Majora-92bf1922-011, 92–96. doi:10.25080/Majora-92bf1922-011
[48] Bingyu Shen, Lili Wei, Chengcheng Xiang, Yudong Wu, Mingyao Shen, Yuanyuan Zhou, and Xinxin Jin. 2021. Can systems explain permissions better? Understanding users' misperceptions under smartphone runtime permission model. In *30th USENIX Security Symposium (USENIX Security 21)*. USENIX Association, San Francisco, CA, USA, 751–768. https://www.usenix.org/conference/usenixsecurity21/presentation/shen-bingyu
[49] Michael Specter, Abbie Farr, Bo Ma, Robin Lassonde, and Mihai Christodorescu. 2025. Fingerprinting SDKs for mobile apps and where to find them: Understanding the market for device fingerprinting. In *Proceedings of the 32nd ACM Conference on Computer and Communications Security (CCS 2025)*. ACM, Taipei, Taiwan.
[50] Mohammad Tahaei, Ruba Abu-Salma, and Awais Rashid. 2023. Stuck in the permissions with you: Developer & end-user perspectives on app permissions & their privacy ramifications. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems* (Hamburg, Germany) *(CHI '23)*. Association for Computing Machinery, New York, NY, USA, Article 168, 24 pages. doi:10.1145/3544548.3581060
[51] Mohammad Tahaei, Ruba Abu-Salma, and Awais Rashid. 2023. Stuck in the permissions with you: Developer & end-user perspectives on app permissions & their privacy ramifications. In *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. 1–24.
[52] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. 2021. Privacy champions in software teams: Understanding their motivations, strategies, and challenges. In

Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 693, 15 pages. doi:10.1145/3411764.3445768

[53] Mohammad Tahaei and Kami Vaniea. 2019. A survey on developer-centred security. In *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE, Stockholm, Sweden, 129–138. doi:10.1109/EuroSPW.2019.00021

[54] Mohammad Tahaei, Kami Vaniea, and Naomi Saphra. 2020. Understanding privacy-related questions on stack overflow. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems* (Honolulu, HI, USA) *(CHI '20)*. Association for Computing Machinery, New York, NY, USA, 1–14. doi:10.1145/3313831.3376768

[55] Joshua Tan, Khanh Nguyen, Michael Theodorides, Heidi Negrón-Arroyo, Christopher Thompson, Serge Egelman, and David Wagner. 2014. The effect of developer-specified explanations for permission requests on smartphone user behavior. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 91–100. doi:10.1145/2556288.2557400

[56] Güliz Seray Tuncay. 2024. Android Permissions: Evolution, Attacks, and Best Practices. *IEEE Security & Privacy* (2024).

[57] Nils Wernerfelt, Anna Tuchman, Bradley Shapiro, and Robert Moakler. 2024. *Estimating the Value of Offsite Tracking Data to Advertisers: Evidence from Meta*. NBER Working Papers 32765. National Bureau of Economic Research, Inc. doi:10.3386/w32765

[58] Wenjia Wu, Jianan Wu, Yanhao Wang, Zhen Ling, and Ming Yang. 2016. Efficient fingerprinting-based Android device identification with zero-permission identifiers. *IEEE Access* 4 (2016), 8073–8083. doi:10.1109/ACCESS.2016.2626395

Appendix

A Survey and sample details

The text for the survey questions and responses are provided in full at the end of the Appendix. A survey pilot was first conducted with 10 participants from the same participant pool as the main sample. We used the pilot to confirm the survey functioned as intended and then made no changes to the survey. Since the pilot survey and participant pool are the same as the expanded sample, pilot responses are included into the full analysis sample.

Table 2 shows the distribution of the sample's age and gender demographics and Table 4 shows the distribution of countries where the developers say they live. In addition to these basic demographics, the survey also asked participants about their roles as developers, including whether they primarily work on an app or SDK (and if they work on multiple, to answer questions for the one they spend the most time on), their development team size, and their years of professional experience as an Android developer. Responses are shown in Table 3.

The survey also asked developers to provide which category either their app or SDK is listed as, depending on whether they said they primarily work on an app or SDK. Response options were from the lists of categories from the Google Play Console [20] and Google SDK Index [17], respectively. Responses are shown in Tables 5 and 6.

B Additional analysis details and robustness checks

Table 9, Figure 8, Table 10, Table 11, Table 12 and Table 13 provide additional data to support the findings presented in Section 4.3. Table 14, Table 15 and Table 16 provide additional data to support the findings presented in Section 4.1.

Table 2: Sample gender and age demographics (N=246)

	n	%
Gender		
Man	214	87.0
Woman	23	9.3
Prefer not to answer	7	2.8
Other	2	0.8
Age		
18 - 24	3	1.2
25 - 34	55	22.4
35 - 44	92	37.4
45 - 54	66	26.8
55 - 64	20	8.1
65 or older	7	2.8
Prefer not to answer	3	1.2

Table 3: Sample developer details (N=246)

	n	%
App or SDK		
App	228	92.7
SDK	18	7.3
Team size		
1	91	37.0
2 - 4	90	36.6
5 - 9	41	16.7
10 - 24	17	6.9
25 - 49	4	1.6
50 or more	3	1.2
Experience		
Less than 1 yr	11	4.5
1 to 2 yrs	26	10.6
3 to 5 yrs	49	19.9
6 to 9 yrs	63	25.6
10 to 14 yrs	69	28.0
15+ yrs	28	11.4

Table 17 provides results for a robustness check for the logistic regression model presented in Table 1. Responses where the participant answered "Yes, with the optional model" are filtered out of this analysis in order to compare the responses for "Yes, with the required mode" to "No, not at all." The results are directionally consistent with the main model (Table 1), yet the effect sizes are even larger. Developers whose apps/SDKs use fingerprinting are even more likely (OR = 8.3 versus OR = 6.5) to support API Usage Purposes with the required model.

Table 18 provides results for another robustness check for the logistic regression model presented in Table 1, where demographic details are included. We did not find demographic variables were statistically significant, whether or not they were aggregated. For simplicity, we report the model with aggregated variables: gender variables are consolidated to male versus non-male (woman/prefer not to say/other), age variables are consolidated to 3 groups (18 - 34 years, 35 - 54 years, 55 or older), participants who did not answer age are excluded (n=3), team size is consolidated to 3 groups (1, 2 -

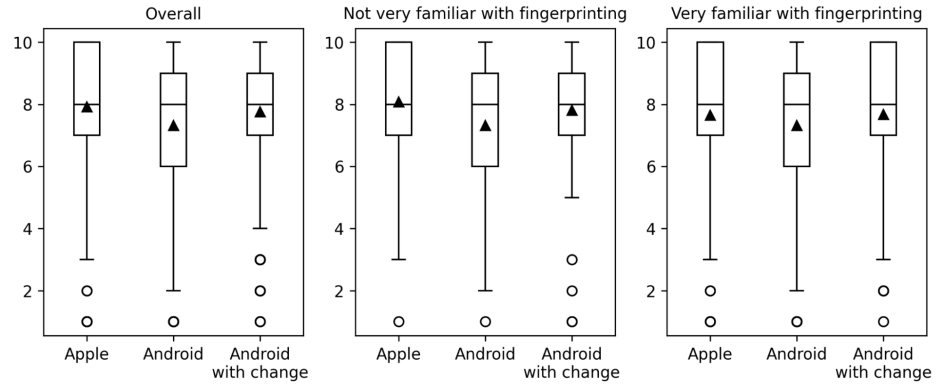


Figure 8: Distribution of levels of agreement for the statements "[Apple/Android] protects user privacy" from 1 (strongly disagree) to 10 (strongly agree) comparing participants overall (left), versus those who did not say they are very familiar with fingerprinting (middle) and those who did say they are very familiar with fingerprinting (right). Lines indicate the 25th, 50th (median) and 75th percentiles; triangles indicate the mean.

Table 4: Sample geographic distribution (N=246)

Country		%
United States of America	88	35.8
United Kingdom	42	17.1
India	33	13.4
Canada	17	6.9
Germany	16	6.5
Finland	6	2.4
Netherlands	6	2.4
Kenya	4	1.6
Malaysia	3	1.2
Italy	3	1.2
Brazil	2	0.8
Nigeria	2	0.8
Sweden	2	0.8
Ireland	1	0.4
Serbia	1	0.4
Indonesia	1	0.4
Singapore	1	0.4
Australia	1	0.4
Trinidad and Tobago	1	0.4
Spain	1	0.4

9, 10 or more) and years of experience is consolidated to 3 groups (Less than 3 years, 3 - 9 years, 10 or more years).

Table 5: Participants' primary app category listing and fingerprinting use (N=228).

	(Q14) Does your app fingerprint users?				Total	
(Q6a) App Category	Yes	Dependency does	No	Not sure	n	%
Productivity	4	4	20	0	28	12.3
Games	5	5	16	1	27	11.8
Tools	1	4	18	1	24	10.5
Education	1	6	14	1	22	9.6
Business	6	4	12	0	22	9.6
Entertainment	2	0	13	2	17	7.5
Finance	5	3	2	0	10	4.4
Travel and Local	2	2	5	1	10	4.4
Lifestyle	0	4	5	0	9	3.9
Shopping	3	1	5	0	9	3.9
Health and Fitness	0	0	8	0	8	3.5
Maps and Navigation	0	1	4	1	6	2.6
Food and Drink	1	1	3	0	5	2.2
Music and Audio	0	2	2	1	5	2.2
Communications	1	0	3	0	4	1.8
Sports	0	1	1	1	3	1.3
Medical	0	0	3	0	3	1.3
Books and Reference	0	1	2	0	3	1.3
Auto and Vehicles	0	1	1	0	2	0.9
Photography	1	0	1	0	2	0.9
Weather	0	0	2	0	2	0.9
House and Home	1	0	0	1	2	0.9
Comics	1	0	0	0	1	0.4
Personalization	0	0	1	0	1	0.4
News and Magazines	0	0	0	1	1	0.4
Social	1	0	0	0	1	0.4
Video Players and Editors	0	1	0	0	1	0.4

Table 6: Participants' primary SDK category listing and fingerprinting use (N=18).

	Does your SDK fingerprint users?				Total	
(Q6b) SDK Category	Yes	Dependency does	No	Not sure	n	%
Data management	3	0	3	0	6	33.3
Analytics	0	0	3	0	3	16.7
Advertising and monetization	1	0	1	0	2	11.1
Marketing and engagement	1	1	0	0	2	11.1
Payments	1	1	0	0	2	11.1
User support	0	1	1	0	2	11.1
User authentication	1	0	0	0	1	5.6

Table 7: Developer participants' relationships to fingerprinting.

	Total		SDK		App	
Question and response	n	%	n	%	n	%
(Q13) "Were you already familiar with device fingerprinting?"						
Very familiar	91	37	12	66.7	79	34.6
Somewhat familiar	123	50	5	27.8	118	51.8
I was not familiar with device fingerprinting, but now I understand it	26	10.6	1	5.6	25	11
Even after the explanation, I do not understand device fingerprinting	6	2.4	0	0	6	2.6
(Q14) "Does your app/SDK fingerprint users"						
Yes	42	17.1	7	38.9	35	15.4
Not directly, but a dependency does	44	17.9	3	16.7	41	18
No	149	60.6	8	44.4	141	61.8
I'm not sure	11	4.5	0	0	11	4.8

Table 8: Developers participants' familiarity with fingerprinting by their app/SDK use.

	(Q14) "Does your app/SDK fingerprint users?"				Total	
(Q13) Familiarity with fingerprinting	Yes	Dependency does	No	Not sure	n	%
Even after the explanation, I do not understand device fingerprinting	2	0	3	1	6	2.4
I was not familiar with device fingerprinting, but now I understand it	0	1	21	4	26	10.6
Somewhat familiar	9	23	85	6	123	50.0
Very familiar	31	20	40	0	91	37.0

Table 9: Participant-level comparisons for levels of agreement to whether "[Apple/Android] protects user privacy" (Q12), by whether or not they are very familiar with fingerprinting (Q13). The data corresponds to Figure 7.

	All		Very familiar		Not very familiar	
	n	%	n	%	n	%
Apple above Android	95	38.6	27	29.7	68	43.9
Equivalent	113	45.9	44	48.4	69	44.5
Android above Apple	38	15.4	20	22.0	18	11.6

Table 10: Distribution of levels of agreement for the statements "[Apple/Android] protects user privacy" (Q12) from 1 (strongly disagree) to 10 (strongly agree).

	All (N=246)			Not very familiar with fingerprinting (N=155)			Very familiar with fingerprinting (N=91)		
	Apple	Android	Android with change	Apple	Android	Android with change	Apple	Android	Android with change
Mean	7.9	7.3	7.8	8.1	7.3	7.8	7.7	7.3	7.7
Std. dev.	2.1	2.2	2.0	1.8	2.0	1.8	2.4	2.5	2.4
Min	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
25%	7.0	6.0	7.0	7.0	6.0	7.0	7.0	6.0	7.0
50%	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0	8.0
75%	10.0	9.0	9.0	10.0	9.0	9.0	10.0	9.0	10.0
Max	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0	10.0

Table 11: Logistic regression model where the dependent variable is whether the developer ranked Apple above Android with respect to protecting user privacy.

Independent variable	Coef.	OR	95% CI for OR	p-value
Intercept	-0.246	0.782	[0.569, 1.073]	0.128
Very familiar with fingerprinting	-0.617*	0.540	[0.311, 0.936]	0.028
N	246			
Pseudo R-squared	0.01512			

Table 12: Robustness check for the logistic regression model where the dependent variable is whether the developer ranked Apple above Android with respect to protecting user privacy.

Independent variable	Coef.	OR	95% CI for OR	p-value
Intercept	-1.632	0.196	[0.037, 1.028]	0.054
App or SDK (Ref: SDK)				
App	1.298	3.662	[0.988, 13.582]	0.052
Team size (Ref: 1)				
2 - 9	0.605	1.831	[0.996, 3.365]	0.052
10 or more	0.288	1.334	[0.485, 3.670]	0.577
Years experience (Ref: Less than 3 years)				
3 - 9 years	-0.087	0.916	[0.409, 2.054]	0.832
10+ years	-0.272	0.762	[0.314, 1.848]	0.547
Age (Ref: 18 - 34)				
35 - 54	0.04	1.040	[0.523, 2.071]	0.910
55+	-0.321	0.725	[0.256, 2.057]	0.546
Gender (Ref: Non-male)				
Male	-0.035	0.966	[0.422, 2.212]	0.935
Very familiar with fingerprinting	-0.626*	0.535	[0.297, 0.962]	0.037
N	243			
Pseudo R-squared	0.04270			

Table 13: Logistic regression model where the dependent variable is whether the developer increased their agreement with the statement that Android protects user privacy after supposing API Usage Purposes are implemented.

Independent variable	Coef.	OR	95% CI for OR	p-value
Intercept	-0.405*	0.667	[0.483, 0.919]	0.013
Very familiar with fingerprinting	-0.304	0.738	[0.429, 1.269]	0.272
N	246			
Pseudo R-squared	0.003754			

Table 14: Responses to survey question Q18a: "What level of developer effort would be required to support API Usage Purposes?"

Level		All	By support response		
			No	Yes, required	Yes, optional
1	Very little effort	28	2	19	7
2	Little effort	80	4	34	42
3	Moderate level of effort	95	14	32	49
4	Significant level of effort	35	5	13	17
5	Very significant level of effort	8	1	4	3

Table 15: Responses to survey question Q18b: "What impact, if any, would this change have on user privacy?"

Level		All	By support response		
			No	Yes, required	Yes, optional
1	It would have a large negative impact on user privacy	4	2	1	1
2	It would have a small negative impact on user privacy	10	1	3	6
3	It would have no impact on user privacy	49	17	9	23
4	It would have a small positive impact on user privacy	130	6	60	64
5	It would have a large positive impact on user privacy	53	0	29	24

Table 16: Responses to whether Android should implement "API Usage Purposes" (Q19), broken down by whether or not the developer works on an App/SDK that fingerprints users (Q14).

	Total		Does fingerprint		Does not fingerprint	
	n	%	n	%	n	%
No, not at all	26	10.6	3	3.5	23	14.4
Yes, optional	118	48.0	49	57.0	69	43.1
Yes, required	102	41.5	34	39.5	68	42.5

Table 17: Robustness check for logistic regression model where the dependent variable is whether the developer said "Yes, with the required model" to whether they think Android should implement the change. Responses for "Yes, with the optional model" are not included.

Independent variable	Coef.	OR	95% CI for OR	p-value
Intercept	-2.421	0.089	[0.005, 1.591]	0.100
Effort level	-1.053**	0.349	[0.177, 0.687]	0.002
Privacy impact	1.715***	5.556	[2.651, 11.643]	<0.001
Does fingerprinting	2.115*	8.288	[1.594, 43.088]	0.012
N	128			
Pseudo R-squared	0.3648			

Table 18: Robustness check for logistic regression model where the dependent variable is whether the developer said "Yes" to whether they think Android should implement the change, where additional demographic variables are included.

Independent variable	Coef.	OR	95% CI	p-value
Intercept	-2.102	0.122	[0.003, 4.961]	0.266
App or SDK (Ref: SDK)				
App	-1.391	0.249	[0.022, 2.781]	0.259
Team size (Ref: 1)				
2 - 9	-0.337	0.714	[0.217, 2.348]	0.579
10 or more	-0.588	0.556	[0.079, 3.917]	0.555
Experience (Ref: Under 3 years)				
3 - 9 years	-1.133	0.322	[0.034, 3.063]	0.324
10+ years	-1.772	0.170	[0.017, 1.724]	0.134
Age (Ref: 18 - 34)				
35 - 54	-1.577	0.207	[0.021, 2.076]	0.180
55+	-2.537	0.079	[0.006, 1.118]	0.060
Gender (Ref: Non-male)				
Male	1.534	4.639	[0.949, 22.683]	0.058
Effort level	-0.834**	0.434	[0.245, 0.770]	0.004
Privacy impact	1.654***	5.230	[2.586, 10.580]	<0.001
Does fingerprinting	1.851*	6.368	[1.253, 32.358]	0.026
N	243			
Pseudo R-squared	0.3571			

Table 19: Themes, descriptions and occurrences for open-ended responses regarding concerns for API Usage Purposes (Q17a).

	Theme	Description	*Example from data*	Total count	Does fingerprint	Does not fingerprint
1	Compliance and enforcement	Developers may lie or evade the policy; Accuracy of detection; Google may not validate or enforce whether the purpose label is properly used; Enforcement process may be arbitrary.	*"I don't see an enforcement mechanism."*	34	9	25
2	Documentation	Clear documentation of requirements and use.	*"I do like very much that the permission can be declared for finger printing or not. Besides that, good documentation of how to implement this changes is paramount."*	5	2	3
3	User experience	Related to user experience; information shown to users; users misunderstanding the information; information/decision overload for users	*"Adding extra layers of information could lead to overwhelming the user."*	15	6	9
4	Developer and engineering challenges	Limits legitimate uses of impacted APIs; Adds to developer workload; Breaks apps or harms backwards compatibility; etc.	*"My major concern with it is the technical complexity of implementing API usage purposes, "*	26	12	14
5	Other		*"I think it's a nice improvement. I use AdMob, though, so I wonder how it would affect my advertising revenue."*	51	28	23
6	No concern	Response blank or response does not include concerns about API usage purposes.	*"None, looks declarative and straight forward"*	115	14	39

C Developer Survey

This section presents the full list of questions and response options used in the developer survey. Data were collected via a Qualtrics survey. Response options were single-select form fields unless otherwise indicated. When the text shows "app/SDK", the survey actually shows either "app" or "SDK" depending on the response to Q3. Horizontal lines indicate page breaks. To avoid answer order bias, response options were randomly ordered when appropriate and Likert scale options were randomly reversed.

Q0. [Consent.]
Your responses to this survey may be used in a research publication. All responses will be anonymized and any reports and presentations about the findings from this survey will not include any information that could identify you.
By continuing in this survey, you acknowledge you are at least 18 years of age and consent to the use of your responses in research publications.

- □ Continue
- □ Exit

Q1. [Multi-select with randomized response option order, excluding last option, "No".]
If the last option is selected, then the developer exits the survey.
Are you a software developer/engineer working on an Android application (app) or software development kit (SDK)?

- □ Yes, I am a software developer/engineer working on an **Android app**
- □ Yes, I am a software developer/engineer working on an **Android SDK**
- □ No, I am not a software developer/engineer working on an Android app or SDK

Q2. [Randomized response option order, excluding last option, "I don't know".]
This is a soft-screener question. Participants who answered incorrectly or "I don't know" were filtered out of the sample. The correct answer is "In the AndroidManifest.xml file using uses-permission tags".
If an Android application needs access to potentially sensitive user data or system features (like location, camera, or contacts), where must the intent to use these permissions typically be declared?

- □ Explicitly requested only in the Java/Kotlin code using a runtime permission check.
- □ Within the project's .gitignore file to prevent accidental exposure.
- □ In the application's build.gradle file, under defaultConfig.
- □ In the AndroidManifest.xm file using uses-permission tags.
- □ I don't know.

Q3. [Randomized response option order.]
The following questions ask about the Android app or SDK you work on. If you work on multiple, please answer for the one you spend the most time on.
Do you primarily work on an app, or on an SDK?

- □ App
- □ SDK

Q4.
How many Android developers work on your App/SDK? If you work on more than one App/SDK, please answer for your primary App/SDK.

- □ Just myself
- □ 2-4
- □ 5-9
- □ 10-24
- □ 25-49
- □ 50 or more

Q5.[Attention check.]
If the participant does not select "2 to 5" then they exit the survey.
This is an attention check. Select 2 to 5.

- □ Just 1
- □ 2 to 5
- □ 6 to 10
- □ More than 10

Q6a.[Only shown if Q3 response was "App"; Dropdown using list from Google Play store]
Which of the following categories is your app listed as?

Q6b. [Only shown if Q3 response was "SDK"; Options shown from Google Play SDK index categories.]
Which of the following categories is your SDK listed as?

Q7.
Which of the following applies to the app/SDK you work on?

- □ Commercial / revenue generating
- □ Open Source
- □ Internal or enterprise use only
- □ Other (please specify):

Q8.
How many **years of professional experience** do you have as an Android developer?

- □ Less than 1 year
- □ 1 to 2 years
- □ 3 to 5 years
- □ 6 to 9 years
- □ 10 to 14 years
- □ 15+ years

Q9.
What is your age in years?

- □ 18-24
- □ 25-34
- □ 35-44
- □ 45-54
- □ 55-64
- □ 65 or older
- □ Prefer not to say

Q10.
What is your gender?

- ☐ Woman
- ☐ Man
- ☐ Other
- ☐ Prefer not to say

Q11. [Optional; Drop down list of countries]
Please specify the country where you live.

Q12. [Random ordering of the Android and Apple options. Likert scale options appeared horizontally in the survey.]

On a scale of 1 to 10 how much do you agree vs disagree with the following statements?

Android protects privacy	**Apple** protects privacy
1 - Strongly disagree	1 - Strongly disagree
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10 - Strongly agree	10 - Strongly agree

Device fingerprinting is a method to identify and track users. It involves:

(a) Collecting device-specific information from APIs
(b) Combining that information to create a unique "fingerprint" to identify the device

Q13. [Response order randomly reversed.]
Were you already familiar with device fingerprinting?

- ☐ Very familiar
- ☐ Somewhat familiar
- ☐ I was not familiar with device fingerprinting, but now I understand it
- ☐ Even after the explanation, I do not understand device fingerprinting

---

Q14. [Response order randomly reversed.]
Does your app/SDK fingerprint users? *Your answers are confidential. We will not attempt to connect your responses to you or your app/SDK.*

- ☐ Yes
- ☐ Not directly, but a dependency does
- ☐ No
- ☐ I'm not sure

Q15.
The following questions explore a **hypothetical change to Android** called "API Usage Purposes".

This change is designed to protect users from unwanted fingerprinting, to improve user privacy.
With this change, Android developers would need to declare their purposes for specific APIs that can be used for fingerprinting.
The specific APIs would include the following.
Does your app/SDK use these? Please mark all included in your app/SDK.

API	Yes	Dep.	No	Unsure
Settings.Secure.ANDROID_ID	☐	☐	☐	☐
InputMethodManager. getEnabledInputMethodList	☐	☐	☐	☐
InputMethodManager. getEnabledInputMethodSubtypeList	☐	☐	☐	☐
InputMethodManager. getInputMethodList	☐	☐	☐	☐
TrafficStats.getTotalRxBytes	☐	☐	☐	☐
TrafficStats.getTotalRxPackets	☐	☐	☐	☐
TrafficStats.getTotalTxBytes	☐	☐	☐	☐
TrafficStats.getTotalTxPackets	☐	☐	☐	☐
AudioManager.getStreamVolume	☐	☐	☐	☐
AudioManager.getStreamMaxVolume	☐	☐	☐	☐
TelephonyManager.getSubscriberId	☐	☐	☐	☐
TelephonyManager. getNetworkCountryIso	☐	☐	☐	☐

API Usage Purposes are designed to protect users from unwanted fingerprinting, to improve user privacy.
Android developers would add "purposes" to `<uses-permission>` xml elements in their `AndroidManifest.xml` file for specific APIs that can be used for fingerprinting.
Here's an example of how API Usage Purposes would work.
Suppose you are the developer for an app/SDK that uses the APIs ⟨*API_1*⟩ and ⟨*API_2*⟩ from the previous table.
Assume there will be new permissions for each of these APIs.
Your `AndroidManifest.xml` file would include the following:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.my.example.app">

    <uses-permission
            android:name="android.permission.PERMISSION_FOR_API_1">
        <purpose android:name="NotForFingerprinting"/>
    </uses-permission>

    <uses-permission
            android:name="android.permission.PERMISSION_FOR_API_2">
        <purpose android:name="NotForFingerprinting" />
    </uses-permission>
    ...
</manifest>
```

Assume tools within Android Studio will help developers fill in the syntax for each API with automatic suggestions.

Q16. [Likert scale options appeared horizontally in the survey.]
Suppose Android required API Usage Purposes for APIs that can be used for fingerprinting.
Given this change, on a scale of 1 to 10 how much do you agree vs disagree with the below statement?

Android protects user privacy
1 - Strongly disagree
2
3
4
5
6
7
8
9
10 - Strongly agree

[The order of the following two questions is randomized.]

Q17a. [Optional; Open ended.]
What concerns, if any, do you have with API Usage Purposes?

Q17b. [Optional; Open ended.]

What benefits, if any, do you see with API Usage Purposes?

Q18a-b. [The order of the following two questions is randomized; the order of response options is randomly reversed.]
Suppose Android required API Usage Purposes as described in this survey.
What level of developer effort would be required to support API Usage Purposes?

- □ Very little effort
- □ Little effort
- □ Moderate level of effort
- □ Significant level of effort
- □ Very significant level of effort

What impact, if any, would this change have on user privacy?

- □ It would have **a large negative impact** on user privacy
- □ It would have **a small negative impact** on user privacy
- □ It would have **no impact** on user privacy
- □ It would have **a small positive impact** on user privacy
- □ It would have **a large positive impact** on user privacy

---

Q19.
Consider two ways Android could implement API Usage Purposes.
Optional model: API Usage Purposes are optional.
Apps that implement API Usage Purposes for all related APIs receive a privacy badge visible to users in the Google Play store and are ranked higher.
Required model: API Usage Purposes are required.
API calls will fail when API Usage Purposes are not provided for the impacted APIs.

Do you think Android should implement API Usage Purposes?

- □ Yes, with the **optional model**
- □ Yes, with the **required model**
- □ No, not at all

Q20. [Optional; Open ended.]
Is there anything additional you'd like to share with Google about this hypothetical change (optional)?

Q21. [Optional; Open ended.]
Do you have any comments on how to improve this survey (optional)?

Thank you for your time!