Assessing Privacy Compliance Awareness and Practices Among Mobile Third-party Library Developers

Fares F. Alharbi*
Indiana University-Bloomington
Bloomington, Indiana, USA and
Taibah University
Medina, Saudi Arabia
fafaalha@iu.edu

Ece Gumusel*
Rutgers University
New Brunswick, New Jersey, USA
ece.g@rutgers.edu

Luyi Xing
University of Illinois Urbana-Champaign
Champaign, Illinois, USA
lxing2@illinois.edu

Xiaojing Liao
University of Illinois Urbana-Champaign
Champaign, Illinois, USA
xjliao@illinois.edu

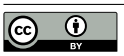
Abstract

Mobile applications increasingly rely on third-party library or Software Development Kits (SDKs) for analytics, advertising, and other core functionalities, yet SDKs often handle privacy-sensitive user data with little guidance on privacy compliance. Despite regulatory frameworks such as GDPR, CCPA, and COPPA, it remains unclear how mobile SDK developers perceive and implement privacy practices. In this paper, we present a mixed-methods study of mobile SDK developers, first surveying 126 participants and then conducting semi-structured interviews with 10 participants to examine awareness of privacy regulations, compliance practices, challenges, and support needs. Our analysis examines the division of compliance responsibilities between SDK developers and downstream mobile app developers, highlighting its implications for mobile supply chain security. We found the following key challenges in compliance practices SDK developers have been facing: (1) too many checkboxes, (2) lack of full transparency in third-party dependencies, and (3) balance between compliance and delivery requirements. Notably, legal professionals and governance, risk, and compliance (GRC) specialists are often found as the primary point of responsibility for these practices even though they are generally courteous with privacy and security risks. We conclude with actionable recommendations to enhance tooling, documentation, and accountability, thereby improving both developer support and privacy enforcement within SDK ecosystems.

CCS Concepts

• **Security and privacy → Human and societal aspects of security and privacy**; **Social aspects of security and privacy**.

*The work was completed with equal contribution, and the first two authors are listed in alphabetical order.



This work is licensed under a Creative Commons Attribution 4.0 International License.
CCS '26, The Hague, Netherlands.
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2871-6/2026/11
https://doi.org/10.1145/3830454.3832742

Keywords

Mobile SDKs; Third-party libraries; Privacy compliance; Mobile supply chain; Usable privacy

ACM Reference Format:
Fares F. Alharbi*, Ece Gumusel*, Luyi Xing, and Xiaojing Liao. 2026. Assessing Privacy Compliance Awareness and Practices Among Mobile Third-party Library Developers. In *Proceedings of the 2026 ACM SIGSAC Conference on Computer and Communications Security (CCS '26), November 15–19, 2026, The Hague, Netherlands.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3830454.3832742

1 Introduction

Mobile Third-Party Libraries or Software Development Kits (SDKs) [1] play a crucial role in the mobile application supply chain. SDKs, integrated by downstream mobile application developers, operate as embedded components within mobile app to add functionality such as analytics, advertising, authentication, and crash reporting. Through these integrations, SDKs can collect, process, and transmit a wide range of sensitive user data and device resource data, sometimes independently of the app's core functionality [22, 31]. Multiple prior studies [31, 48] have shown that this access can be abused to aggregate and exfiltrate personal data. As a result, data flows in mobile apps are shaped not only by mobile app developers as first-party but also by the behavior of third-party SDKs. This tight coupling creates complex dependency relationships between SDK providers and downstream app developers, raising important questions about transparency, accountability, and shared responsibility for data collection and handling within the broader mobile supply chain. Recently, a US court decision (i.e., *Greenley v. Kochava* [5]) has even acknowledged that third-party SDK integrations can enable unauthorized collection and sale of geolocation data, underscoring the legal risks of inadequate disclosures. This increases the responsibility on SDK developers to ensure compliance and adopt strong privacy practices for their app users.

However, a significant gap remains in systematically understanding how mobile SDK developers implement privacy compliance practices and how these efforts are coordinated with downstream mobile application developers. These practices include adoption

[1] In this paper, we use the terms "SDK" and "library" interchangeably.

and usage of tools that help improve SDK compliance (e.g., those detect privacy violations), the provision of SDK-specific disclosure artifacts and privacy related documents for downstream mobile app developers (such as data collection manifests, third-party Libraries' Privacy Label Disclosure Guidelines), the design of data collection mechanism and configuration, user consent signaling and enforcement, etc. Addressing this gap is essential for developing more robust accountability frameworks and establishing best practices for third-party SDK privacy compliance practices within the mobile software supply chain. Note that in this study, we focus on SDK developers' privacy compliance practices rather than evaluating the function-driven data practices of SDKs themselves.

Research Questions. To investigate mobile SDK developers' practices regarding privacy compliance, this empirical study addresses the following research questions:

- **RQ1.** How do mobile SDK developers perceive their responsibilities for privacy compliance and data protection within the mobile software supply chain, and what trade-offs do they consider?
- **RQ2.** How do mobile SDK developers implement privacy-compliant practices, design patterns, and processes, and how is the effectiveness of these practices?
- **RQ3.** What integration challenges do mobile SDK developers face in adopting privacy-relevant SDKs while maintaining privacy compliance, and how are the challenges influenced by their team size?

Study and Findings. In this study, we conducted a mixed-methods investigation combining semi-structured interviews with SDK developers and a survey designed to capture a broad range of perspectives. The survey comprises five sections: (1) demographic information; (2) perceptions of privacy responsibility; (3) current privacy compliance practices; (4) challenges encountered in interpreting regulatory requirements; and (5) developers' needs, expectations, and desired forms of support for facilitating compliance. These components allow us to characterize how SDK developers navigate regulations such as GDPR, CCPA/CPRA, COPPA, and LGPD within real-world development workflows. In addition to the survey, we conducted semi-structured interviews with 10 active SDK developers to complement the quantitative findings and provide deeper insights into their privacy-related experiences.

For **RQ1**, the majority of SDK developers have a moderate to high level of familiarity with privacy regulations. The majority indicated that SDK developers should be responsible for privacy compliance (78.6%). However, in the interviews, participants emphasized that their role primarily involves following standards set by legal, security, or backend teams, rather than determining compliance themselves framing their responsibility as limited to secure engineering practices such as encryption, key management, or data minimization.

Regarding SDK development associated with privacy compliance (**RQ2**), SDK developers incorporate privacy compliance through a combination of early-stage design decisions, established organizational processes, and tool-supported validation workflows, with the most frequently selected points in the lifecycle being the development phase . The most frequently reported practices were the use of compliance frameworks such as GDPR and CCPA checklists and internal privacy reviews . SDK developers predominantly rely on automated tools and manual reviews to evaluate privacy risks, with automated privacy or security scanning . The majority of also respondents considered their processes are very effective .

In addition, our study finds that SDK developers encounter a range of challenges when attempting to achieve privacy compliance (**RQ3**), spanning regulatory complexity, uncertainty introduced by third-party components, limited tooling, and pressure from product timelines. The majority of survey participants reported difficulty keeping up with regulatory compliance updates, such as GDPR (66.9%), while over half noted challenges related to assessing third-party behavior (52.1%), and 47.1% reported encountering a lack of clarity or examples and the burden of excessive manual work due to poor tooling. Balancing privacy compliance with product delivery timelines is perceived as moderately to very challenging by the majority of respondents. Team size significantly influences these challenges: medium-sized teams (6-10 people) had higher odds of reporting implementation challenges , while small teams (1-5 people) were less likely to perceive integration as difficult , and very large teams (20+ people) reported substantially higher odds of perceiving integration difficulty .

Contributions. In this work, we make the following contributions:

- We present the first empirical SDK developers study on understanding SDK developers privacy compliance practices with a focus on the role of their SDK adoption experience.
- We show that among SDK developers, privacy compliance awareness often exists without actionable understanding. This study also identified several challenges in the development of privacy-compliant SDKs, including regulatory complexity, uncertainty introduced by third-party components, limited tooling, and pressure from product timelines.
- This study provides recommendations for privacy-compliant Mobile SDK development.

Artifact Disclosure. Artifacts supporting this research, including recruitment materials, survey and interview questionnaires, and the codebook, are provided in our supplementary material.

2 Background

2.1 Privacy Obligation to SDKs

The extensive use of third-party libraries in modern software development has raised significant data privacy concerns, as these components may collect, process, or transmit personal data beyond the direct control of application developers. Common examples include analytics libraries that collect IP addresses, device identifiers, and location information [9, 39], crash-reporting SDKs that transmit user interaction data [37], advertising SDKs that track user behavior and build profiles [9]. Such practices bring SDK directly within the scope of contemporary data protection regulations.

Under the General Data Protection Regulation (GDPR), third-party libraries that process personal data on behalf of an organization may be classified as *data processors*. Article 28 of the GDPR requires data controllers to ensure that processors provide sufficient guarantees to implement appropriate technical and organizational measures [46]. In addition, Article 25 mandates data protection by design and by default, obligating organizations to configure third-party SDK to minimize personal data collection [12, 16]. Article 44 further regulates international data transfers [19], which is particularly relevant when third-party libraries transmit data to servers

located outside the European Union. Similarly, the California Consumer Privacy Act (CCPA) imposes obligations on organizations that share personal data with third parties, including software library providers. The CCPA defines personal information broadly to include identifiers such as IP addresses and online activity data, which are commonly collected by embedded SDK [10, 48]. Section 1798.140(ag) distinguishes service providers from third parties and restricts the use of personal data for purposes beyond those specified in the contract [36]. For example, an analytics library configured to reuse collected data for its own commercial purposes may trigger non-compliance.

These regulatory requirements illustrate that third-party SDKs are no longer only technical components but legally significant actors within sensitive data processing ecosystems. In this study, we investigate this convergence of legal accountability and mobile software dependency management.

2.2 Mobile SDK Data Practices

SDKs collect various types of user data . For some SDK categories, privacy-relevant data collection is not incidental but central to the service provided. Analytics SDKs typically gather behavioral data including user actions, session duration, and navigation patterns. Advertising SDKs collect device identifiers (e.g., IMEI, Android ID, advertising IDs), location information, and detailed user profiles for targeted advertising. Social media SDKs may access authentication credentials, profile information, and social graph data, while payment SDKs handle financial information and transaction records. Meanwhile, many SDKs such as those featuring developer utilities or crash reporters, expose privacy-relevant data through improper logging, permissions, or configuration practices. In such cases, privacy challenges often arise less from the SDK's core functionality but more from SDK developers' implementation practices related to engineering, documentation, compliance understanding and tooling, development support resources, etc.

A critical challenge in the SDK ecosystem is the lack of transparency in data practices. Research has shown that SDKs can engage in unexpected data collection behaviors, including cross-library data harvesting where one SDK accesses data collected by another SDK within the same application [48]. Even benign SDKs may collect more data than documented or share data in ways not clearly disclosed to users or app developers. The mobile software supply chain structure further complicates privacy compliance. Unlike first-party app code where developers maintain full control, SDKs introduce privacy-relevant third-party components with their own data practices, update cycles, and compliance requirements. This distributed responsibility, spanning SDK developers, app developers, and platform providers, creates uncertainty about accountability for data processing activities, as regulations such as GDPR and CCPA require clear identification of data controllers and processors throughout the entire data lifecycle.

SDK Developers' Role in Privacy Compliance. In the regulatory landscape of this study, developers are expected to maintain transparency regarding the types of user privacy-sensitive data they collect, the third parties with whom this data is shared, and the purposes of data processing, primarily communicated through privacy labels [1–4]. Yet, developers struggle to maintain accurate privacy labels in practice, particularly due to the frequent changes and updates in third-party SDKs [27, 28, 51]. Although some third-party SDK providers have issued guidelines detailing the data collection and processes disclosures developers should make when integrating the SDKs, these guidelines are also often incomplete or inaccurate, even among widely used SDKs, resulting in non-compliant privacy labels and inconsistencies between declared and actual data practices [23, 49].

2.3 Study Scope

This study focuses on understanding privacy compliance practices among mobile SDK developers working across iOS, Android, and cross-platform development environments. Our investigation encompasses SDK developers' awareness, perceived responsibilities, implementation practices, and challenges related to privacy compliance broadly, rather than limiting our focus to specific regulations.

Geographic Scope. All survey participants ($N = 126$) are based in the United States. Our interview participants ($N = 10$) represent a geographically diverse sample: all were residing in the U.S. at the time of the study, though their backgrounds reflect varied international experience: six participants are from the United States, two from Germany, one from India, and one from Egypt. This geographic distribution reflects the global nature of mobile SDK development, where developers from different jurisdictions contribute to SDKs that may be deployed worldwide.

Regulatory Scope. While we investigate privacy compliance broadly, our study specifically examines developers' familiarity with and practices related to major privacy regulations including GDPR (European Union), CCPA/CPRA (California, USA), COPPA (Children's Privacy, USA), HIPAA (USA), and LGPD (Brazil). Participants were also given the option to identify other relevant regulations beyond these commonly recognized frameworks. This approach allows us to capture both the most prevalent regulatory concerns and emerging or region-specific compliance requirements that SDK developers encounter in practice.

Platform and SDK Type Scope. Our study encompasses SDKs developed for iOS, Android, and cross-platform environments, reflecting the diverse mobile software ecosystem. We include various SDK types such as analytics/tracking, advertising/monetization, social/sharing features, authentication/security, developer utilities, and payments/finance SDKs. This broad platform and functional scope enables us to identify compliance patterns and challenges across the mobile software supply chain.

Study Limitations. Our study does not include technical evaluation of SDK implementations, static or dynamic code analysis, behavioral measurements, or assessments of regulatory enforcement. Instead, we focus on understanding the human factors—specifically, SDK developers' perceptions, practices, and challenges in navigating privacy compliance requirements within their development workflows.

3 Methodology

To address our research questions, we conducted a within-subject online survey, followed by semi-structured interviews with SDK developers. This section presents our empirical study design, which

investigates SDK developers' awareness of privacy compliance, their practices, and the challenges they face.

3.1 Online Survey

We conducted a within-subject survey using Qualtrics[2]. The survey was developed by researchers from diverse disciplines and backgrounds in the U.S., including security, privacy, and law and policy. After obtaining consent and confirming participants' prior experience with SDK development, the survey was structured into five parts. Part I collected participants' background information, including demographics and current SDK development practices. Part II measured participants' awareness of regulatory privacy compliance. Part III focused on their privacy compliance practices and processes in SDK development. Part IV examined the difficulties and obstacles participants encounter in implementing privacy practices. Finally, Part V assessed participants' needs for resources, tools, or support to address privacy compliance challenges. These sections were assessed using questions that participants answered on a five-point Likert scale, as well as multiple-choice items. The complete set of survey questions can be found in §A.2 of the supplementary material.

3.1.1 Pilot Study and Reliability Check. Before distributing the survey, we conducted a pilot study to evaluate the research procedures, estimate the average survey completion time, and assess the reliability and consistency of the survey questions. 8 active SDK developers (5 men, 2 women) whose ages ranging from 22 to 45+ participated in the pilot study. They provided detailed feedback on question comprehension, interpretation, and response processes. Based on their input, we refined the wording of questions, scale descriptions, and instructions to ensure clarity and consistency, and shortened the questionnaire. We then performed preliminary statistical analyses, including internal consistency checks and item-level correlations. Following these refinements, we conducted a second pilot with 4 active SDK developers. We assessed their understanding and ability to complete the survey, which took approximately 8 minutes. The results indicated that the majority of participants could successfully comprehend and complete the survey, after which we proceeded with full distribution to the target population.

3.1.2 Recruitment. The survey was conducted between October and November 2025. Using a convenience sampling approach, the researchers first distributed the survey via Prolific[3], then posted recruitment information on various popular social media platforms, including LinkedIn. Later, participants were encouraged to further share the survey through their personal networks. During recruitment, we deliberately avoided mentioning that the study focused on privacy and security to reduce potential sample bias. Participants were required to meet the following criteria: (1) 18 years or older, (2) residing in the U.S., (3) proficient in English, and (4) having prior experience with SDK development. All participants were volunteers and provided informed consent prior to participation. Participants received a compensation of $1 USD for their time upon completion of the survey on Prolific. We made sure the compensation was aligned with the Prolific's policy standards for a survey of this length, which was also approved by our IRB.

3.1.3 Sample Size. Before proceeding with the survey, participants were required to confirm their SDK development experience. The survey also included attention checks (see §A.2 of the supplementary material) to filter out inattentive responses, and consistency filters were applied during data cleaning. Of the total 291 survey respondents, 11 participants failed attention-check questions, 52 did not identify themselves as SDK developers, and 102 did not complete the survey. After removing invalid responses, 126 participants successfully completed the survey, with an average completion time of 7.13 minutes. Our final sample of 126 participants exceeded the required threshold, determined via an a priori power analysis using G*Power (medium effect size, Cohen's $d = 0.5$, $\alpha = 0.05$, power = 0.8; minimum 88 participants).

3.1.4 Survey Participants' Background. The study included 126 SDK developers in the U.S., representing a diverse and experienced population (Table 1). The majority of participants identified as Software Engineers (70.2%), followed by Product Managers (17.4%) and Security Engineers (7.4%), with smaller numbers of Compliance/Legal Specialists and other roles (2.5% each). While 39.7% focused exclusively on Mobile SDK development, the remaining 60.3% reported overlapping responsibilities, including Web Developer, Mobile App Developer, Open-source Contributor, or Privacy Engineer, indicating a multidisciplinary sample.

Participants reported substantial professional experience, with 25.6% having more than 10 years, 24.8% between 5 and 10 years, 20.7% between 3 and 5 years, 22.3% between 1 and 3 years, and 6.6% with less than one year. They developed SDKs for multiple platforms, most commonly Android (68.9%), iOS (64.8%), Web (55.7%), and cross-platform technologies such as React Native (38.5%) and Flutter (37.7%).

The primary SDK types participants worked on were diverse, including analytics and tracking (50.8%), developer utilities (48.4%), UI kits (45.1%), authentication and security (42.6%), payments and finance (40.2%), social and sharing (36.1%), and advertising and monetization (31.1%). Most respondents worked in small to mid-sized teams, with 31.4% in teams of 1–5 people, 37.9% in teams of 6–10, 19.8% in teams of 11–20, 5.8% in teams of more than 20, and 5.8% operating independently. Overall, the sample reflects a broad spectrum of SDK practitioners with multi-platform experience, diverse team structures, and varied SDK types, providing rich insights into the SDK development ecosystem.

3.2 Interview

To explore the study research questions in-depth, we conducted further semi-structured interviews with 10 active SDK developers. The interviews comprised 20 questions covering the similar topics included in the survey, aiming to gather and align insights into developers' perceptions of privacy responsibility, their compliance practices, the challenges they face when interpreting or implementing privacy regulations, and the types of support or tooling they believe would improve their ability to meet privacy requirements.

[2]https://www.qualtrics.com/
[3]https://www.prolific.com/

Table 1: Background of Survey Participants (N = 126)

Primary Role	Professional Responsibilities	Professional Experience	Platforms	Primary SDK Types	Team Size
Software Engineer (88, 70.2%)	Mobile SDK only (50, 39.7%)	>10 years (32, 25.6%)	Android (87, 68.9%)	Analytics/Tracking (64, 50.8%)	1–5 people (40, 31.4%)
Product Manager (22, 17.4%)	Overlapping responsibilities (76, 60.3%)	5–10 years (31, 24.8%)	iOS (82, 64.8%)	Developer Utilities (61, 48.4%)	6–10 people (48, 37.9%)
Security Engineer (9, 7.4%)		3–5 years (26, 20.7%)	Web (70, 55.7%)	UI Kits (57, 45.1%)	11–20 people (25, 19.8%)
Compliance/Legal Specialist (3, 2.5%)		1–3 years (28, 22.3%)	Cross-platform (47, 37.7%)	Authentication/Security (54, 42.6%)	20+ people (7, 5.8%)
Other (3, 2.5%)		<1 year (9, 6.6%)	React Native (49, 38.5%)	Payments/Finance (51, 40.2%)	Independent (7, 5.8%)
			Flutter (47, 37.7%)	Social/Sharing (45, 36.1%)	
				Advertising/Monetization (39, 31.1%)	
				Other (4, 3.3%)	

The interviews took place on Zoom and each interview lasted approximately 20 minutes. The interview questionnaire can be found in §A.3 of the supplementary material.

3.2.1 Recruitment. Participants for the semi-structured interviews were recruited from the online survey respondents who expressed interest in the interview. Using convenience sampling approach, we later posted recruitment information on various popular social media platforms, including LinkedIn and Reddit, and later asked friends and colleagues to further share the interview study through their social networks. This approach allowed us to collect data beyond the researchers' immediate social circles and capture responses from a diverse range of U.S.-based SDK developers. During recruitment, we deliberately avoided mentioning that the study focused on privacy and security to reduce potential sample bias. From a pool of 148 potential candidates, we conducted a total of 10 semi-structured interviews. Note that thematic saturation was reached by the 6th interview, and the study concluded after the 10th, as no new themes emerged. All interviewees met the same eligibility criteria as the survey, including being 18 years or older, residing in the U.S., proficient in English, and having prior experience with SDK development. Participation was voluntary, and informed consent was obtained from all interviewees prior to the sessions. Participants received a compensation of $25 USD after the completion of the interview.

3.2.2 Interview Participants' Background. Table 2 presents the background of the 10 interview participants (N=10). Participants held professional roles primarily focused on the design, development, and maintenance of software development kits (SDKs), including Software Developer, SDK Developer, Senior SDK Developer, SDK Maintainer, and Independent Software Developer.

With respect to key skills and tools, participants reported expertise in mobile and web SDK development (iOS, Android, JavaScript), backend technologies (Python, Java), and integration mechanisms such as REST APIs and OAuth. Several participants highlighted experience with SDK architecture and API design, as well as the use of supporting tools including GitHub, monitoring and debugging utilities, traffic inspection tools, and CI/CD pipelines. Some participants also reported experience working with large language models for scripting and tooling assistance.

The reported work contexts span a variety of application domains and development settings. Participants described building and maintaining SDKs for SaaS platforms, mobile applications, payment systems, and developer tooling platforms, as well as contributing to long-term government projects and open-source libraries. Common activities included integrating third-party SDKs, ensuring compatibility in real-world deployment scenarios, managing dependency conflicts, and adhering to constraints such as privacy-by-design policies and regulated-domain requirements.

Company size varied across participants, ranging from small and medium-sized organizations (201–500 and 1K–5K employees) to large enterprises (5K–10K and 10K+ employees), as well as open-source, community-maintained projects without a traditional organizational structure. Company size was determined using publicly available information at the time of the study.

Finally, participants reported between approximately 6 and 13 years of professional experience. Most participants had at least six years of experience, with several reporting a decade or more of involvement in software and SDK development.

3.3 Data Analysis

3.3.1 Online Survey. We first descriptively analyzed means, standard deviations, and distributions for each privacy-related item in Parts II to V of the survey. Next, we examined the influence of demographic factors, as well as SDK role and experience, on privacy perceptions using generalized linear models (GLMs). Five-point Likert scale items were treated as continuous dependent variables, while multiple-choice items were treated as categorical variables. Both these items and the categorical predictors (demographics, SDK role, and experience) were dummy-coded. Predictor significance was evaluated using Wald χ^2 tests, with 95% confidence intervals computed for all coefficients. Interaction terms between demographics and SDK role and experience were also explored to assess whether the effect of SDK role on privacy compliance attitudes varied across subgroups. This approach allowed us to quantify both the independent and interactive effects of demographics and SDK experience on multiple dimensions of privacy compliance knowledge and practices. Lastly, free-text responses were inductively coded using thematic analysis [11]. A primary researcher developed a codebook from all responses, which a second researcher then used to independently code the data. Inter-coder reliability was high (Cohen's $\kappa > 0.81$), and overarching themes were identified from the final codes.

3.3.2 Semi-structured interviews. After cleaning each interview transcript, we analyzed the qualitative data using NVivo R15, applying iterative thematic analysis [11] through two coding cycles [14, 33, 35] : we first applied a deductive approach using codes derived from the survey data, and then refined the codebook inductively based on the interview data through regular research team meetings. More specifically, in the first cycle of coding, two coders

Table 2: Background of Interview Participants (N = 10)

Participant	Role / Job Title	Key Skills / Tools	Work Context	Company Size*	Experience (yrs)
P1	Software Developer	App/SDK development, iOS/Android	Builds mobile apps and SDKs for apartment management companies; handles data for tenants/admins	5K-10K	~ 6 yrs
P2	Software Developer	Python, Android/iOS SDKs, REST APIs, OAuth, WebSockets, GitHub	Develops SDKs for SaaS; manages GitHub repos; integrates third-party APIs	1K-5K	~ 6 yrs
P3	Senior Software Developer	Government project development, Testing, Java	Part of a long-term government project (10 years); Deploys applications to AWS; integrates third-party SDKs; automates pipelines)	10K+	~ 7 yrs
P4	Developer	Mobile SDKs, Privacy-by-design development	Develops and maintains an iOS SDK for a developer tooling platform, Follows a strict no–third-party-dependency policy	201-500	~ 10 yrs
P5	SDK Developer	JavaScript, Charles (Monitoring traffic)	Works on payment SDK and card-based payment aggregators	1K-5K	~ 10 yrs
P6	Independent Software Developer	LLMs for scripting and tooling assistance, GitHub, iOS/macOS	Maintains and contributes to open-source libraries	Open-source project (community-maintained)	~ 12 yrs
P7	Senior SDK Developer	Mobile SDK architecture, API design	10 years of experience in the SDK development; works on commercial SDKs in regulated domains (insurtech, telematics)	Open-source project (community-maintained)	~ 6 yrs
P8	SDK Maintainer	GitHub, SDK development	Develop and maintain commercial SDKs; ensures compatibility in real-world integration scenarios.	201-500; 501–1K (worked at two companies)	~ 8 yrs
P9	SDK Developer	REST API, Python	Develop and maintain a crash-reporting SDK	1K-5K	~ 8 yrs
P10	SDK Developer	Python, OAuth, REST APIs	Builds and maintains SDKs for a SaaS product to allow external developers to integrate with the company's services	501–1K	~ 13 yrs

[4]Company size was determined using publicly available information from company LinkedIn pages at the time of the study.

reviewed transcripts at the sentence level and applied deductive coding aligning with the survey findings to create an initial codebook. In the second cycle, both coders simultaneously conducted inductive coding based on the initial codes and met weekly with the research team, to discuss emerging codes, resolve discrepancies, ensure consistency in coding, and finalize the codebook. Our final codebook (§A.4 of the supplementary material) overall achieved high inter-coder reliability (82% in phase 1 and 97% after consensus-building) and Krippendorff's α (0.82–0.96) with high consistency.

Krippendorff's α was chosen here because it accommodates multiple coders, different measurement levels, and missing data, making it suitable for complex qualitative datasets.

4 Results

Here, we report all our findings regarding our research questions.

4.1 RQ1: SDK Developers' Perceptions of Privacy Compliance

4.1.1 Applied Privacy Regulations for SDKs. SDK developers were asked in the survey to indicate their familiarity with privacy regulations (e.g., GDPR, CCPA, COPPA, etc.). Responses were measured on a 5-point Likert scale. Among the respondents ($N = 126$), 55 developers (43.6%) reported being very familiar, 28 (22.2%) extremely familiar, 26 (20.6%) somewhat familiar, 17 (13.4%) slightly familiar, and 0 not at all familiar with privacy regulations. Our study shows that the majority of SDK developers have a moderate to high level of familiarity with privacy regulations, with over 66% reporting at least being very familiar.

Responses were centered around the upper range of the scale, with a mean of 3.69 and a median of 4 ("Very familiar"). Only a small proportion indicated low familiarity, suggesting that most SDK developers possess a substantial working understanding of privacy-related expectations relevant to their development workflows.

Survey participants were also asked to indicate which privacy regulations are most relevant to their SDKs, allowing multiple selections. Our study shows that the majority of developers identified CCPA/CPRA (California, USA) as relevant (106 respondents; 84.1%), followed by GDPR (EU) (62; 49.2%), COPPA (Children's Privacy, USA) (59; 46.8%), others stated HIPAA (USA) (25; 19.8%), and LGPD (Brazil) (8; 6.3%). These findings indicate that SDK developers consider primarily U.S. and European privacy regulations as most pertinent to their development and compliance efforts.

4.1.2 Confidence in Understanding Privacy Compliance Requirements. In addition, our survey asked SDK developers how confident

they were in their understanding of privacy compliance requirements (e.g., GDPR and CCPA/CPRA). The largest group of respondents reported being very confident (46; 36.5%), followed by those who were moderately confident (39; 30.9%), slightly confident (19; 15.0%), extremely confident (16; 12.6%), and a very small minority reporting not confident (1; 0.8%). Overall, the majority of SDK developers indicated moderate to high confidence in their understanding of privacy compliance requirements, suggesting a generally strong awareness among the surveyed participants. Self-assessed confidence in interpreting privacy compliance requirements also aligns with these findings. Responses were skewed toward higher confidence levels, with a mean confidence score of 3.55 and a median of 4 ("Very confident"). Only a very small number of respondents selected the lowest confidence levels, suggesting that most developers feel reasonably well equipped to meet compliance expectations in practice.

Awareness of emerging data privacy risks. Interview participants expressed awareness of emerging privacy risks, particularly when integrating third-party SDKs, cloud services, or telemetry functionality. One participant noted,

> *"Even [device IP] can be abused…I don't want to store this kind of data. (P6)"*

However, many participants reported that their SDKs do not actively collect sensitive user data, reflecting a limited engagement with broader privacy implications. For instance, one developer noted:

> *"For my personal projects and the open-source SDKs I write, I collect no data; the SDKs provide functionality only, and I have no intention of collecting any user information. (P4)"*

Other participants described their SDKs as primarily handling event-based or operational data rather than regulatory defined personally identifiable information (PII):

> *"The SDK mainly records events like user clicks, card additions, and navigation within the app. Nothing relates to user identity unless consent is explicitly provided. (P8)"*

Similarly, some SDKs were described as collecting only basic usage metrics to understand user engagement or performance:

> *"It collects user IDs, session duration, device type, and in-app actions—just basic data to help track performance and engagement. (P9)"*

4.1.3 Perceived Regulatory Compliance Responsibility. Survey participants were asked to indicate who should be responsible for privacy compliance when an SDK is used in a mobile app. Among the respondents (N = 126), the majority indicated that SDK developers should be responsible (99 respondents; 78.6%), followed by App developers (86; 68.3%), Governance, risk, and compliance teams (64; 50.8%), Privacy teams (59; 46.8%), Platform providers such as Apple or Google (60; 47.6%), and Legal professionals (44; 34.9%).

SDK developers were later asked to indicate the extent to which they feel responsible for ensuring privacy compliance of the SDK.

42 developers (33.3%) reported feeling very responsible, 36 developers (28.5%) moderately responsible, 32 developers (25.4%) fully responsible, and 16 developers (12.7%) slightly responsible. These findings indicate that the majority of SDK developers perceive themselves as accountable for privacy compliance, with nearly 88% reporting at least a moderate level of responsibility, while only a small minority felt slightly responsible.

Compliance responsibility standards set by other teams. Interview participants mostly described privacy compliance as a shared responsibility, relying on legal or compliance teams for guidance. Aligning with the survey, in the interviews, participants emphasized that their role primarily involves following standards set by legal, security, or backend teams, rather than determining compliance themselves. One developer explained,

> *"Another person within the company…there's a department that's gonna check…I just mainly collect and process the data and make sure I'm following the proper standards. (P6)"*

Similarly, another participant noted,

> *"Overall compliance is handled higher up, such as QA on software development timelines. I'm not responsible for deploying the infrastructure for the REST API. I focus on SDK development and application development, but in terms of compliance, other than handling encryption within the app, there's not much I manage directly. (P10)"*

Developers further highlighted that formal compliance responsibility lies with security and governance teams, rather than individual SDK developers:

> *"We have an in-house global security team that figures out the best approaches for compliance and funnels that down to various groups. We are not responsible for 100% compliance, but we enforce rules like separation of data between environments. (P2)"*

Limited compliance and data minimization practices. Developers framed their responsibility as limited to secure engineering practices, such as encryption, key management, or data minimization:

> *"We have an in-house global security team that defines compliance approaches and distributes them to various groups. (P5)"*

Most also described the burden of keeping up with evolving platform and regulatory requirements:

> *"It can be difficult because there are so many things to check, and regulations are also becoming more detailed. (P3)"*

> *"We have an in-house global security team that defines compliance approaches and distributes them to various groups. We are not responsible for 100% compliance, but we enforce rules like separation of data between environments. (P2)"*

4.2 RQ2: Privacy-compliant Practices in Mobile SDK Development

4.2.1 Compliance Practice Stages in SDK Development. The survey results show that SDK developers incorporate privacy compliance through a combination of early-stage design decisions, established

organizational processes, and tool-supported validation workflows. When asked at which stages privacy is considered, the most frequently selected points in the lifecycle were the development phase (75.4%), followed by the design phase (66.4%), testing or QA (57.4%), and post-release maintenance (44.3%). Only a small minority (4.1%) reported not considering privacy at any stage.

During the interviews, we also captured similar patterns in compliance practices, which begin either at the start of development or later during the testing phase.

Privacy compliance in development lifecycle. Participants said that their privacy team is helping their compliance procedures at the beginning of the development while the others expressed they follow the guidelines when testing or launching:

> *"The privacy team is involved from the beginning... they provide guidelines on consent, anonymization, and other requirements. (P9)"*

4.2.2 Practice Steps. According to the survey, SDK developers adopt a combination of structured frameworks, technical safeguards, and internal review processes to maintain privacy compliance. The most frequently reported practice was the use of compliance frameworks, such as GDPR and CCPA checklists, followed by internal privacy reviews, selected by 94 (74.6%) and 87 participants (69.0%), respectively. Other commonly reported practices included data minimization and encryption (77 participants, 61.11%), automated tool analysis (70 participants, 55.6%), managing user consent flows (64 participants, 50.8%), and manual privacy disclosures (63 participants, 50.0%). Only one participant (0.8%) reported "Other" practices.

Beyond the survey responses, developers outlined compliance practices shaped by organizational policies and third-party SDK requirements.

Organizational policy governance in privacy compliance. SDK developers emphasized that organizational policies play a central role in ensuring privacy compliance. Rather than relying solely on individual developers, companies typically implement structured processes to manage SDK usage, monitor third-party vendors, and enforce security standards:

> *"I utilize third-party SDKs following normal company procedures. I check for malicious code, backdoors, or outdated SDKs, but the backend team sets the permission levels. I just ensure that what I use complies with company security procedures. (P10)"*

> *"Another person or department checks compliance. I mainly collect and process data and make sure I follow proper standards so that the back-end team can process it for privacy. (P1)"*

> *"Most major corporations do a good job ensuring you don't just pull in random, unvetted SDKs. There are whitelists and reviews that guide what can and cannot be used. (P2)"*

Reliance on approved third-party SDKs. Another strong theme was reliance on approved SDKs from trusted vendors. Developers are generally prohibited from using unvetted or unauthorized SDKs:

> *"We only use SDKs approved by our company. Otherwise, it could be considered a security or privacy violation, and I would definitely get in trouble. (P10)"*

> *"We have a Tech Catalog listing all approved SDKs. Before we can use an SDK, we must submit a solution review, which the team checks to ensure nothing is missing, then we are allowed to use it. (P5)"*

4.2.3 TDL Privacy Risk Assessment. Survey participants reported that they predominantly rely on automated tools and manual reviews to evaluate the privacy risks of included third-party libraries, although a small subset lacks formal assessment procedures. The most common approach was automated privacy or security scanning, reported by 95 participants (75.4%). Manual code reviews were also widely used, with 72 participants (57.1%) employing this practice. Vendor documentation was referenced by 69 participants (54.8%). A smaller proportion of respondents indicated that they had no formal process for assessing TDL privacy risks (10 participants, 8.0%) or selected "Other" (1 participant, 0.8%).

Exclusion of SDKs with known vulnerabilities. From the interviews, a key emerging theme was that developers exclude SDKs with reported vulnerabilities or data breaches. They reported that SDKs with known issues or past security incidents are removed from the list, preventing their use within the company:

> *"I have used a third-party vendor, but they had a data leak in their procedure. We had to take them off our whitelist of providers, though I didn't experience the actual security violation myself. The company communicated the change down the pipeline to all employees. (P1)"*

Developers also acknowledged that some SDKs may collect more data than documented, requiring careful review of documentation and potential safeguards:

> *"I would investigate network calls, review documentation, and contact the SDK provider if necessary to ensure privacy standards are maintained. (P2)"*

4.2.4 Tool Usage. The survey participants also use a diverse set of tools to support compliance activities. The most commonly reported were platform-provided tools such as the Apple Privacy Manifest and the Google Play Console (64.8%), followed by custom internal tools (41.0%) and third-party compliance tools (37.7%). This combination indicates that SDK teams rely on both ecosystem-mandated mechanisms and organization-specific solutions to maintain compliance.

In addition to the survey, interview participants discussed their governance and compliance approaches.

Using data governance protocols. Developers specifically reported using tools and protocols to minimize data collection:

> *"I checked the privacy manifest of my own SDK. For the SDKs I work on, I enable sensitive permissions carefully—for example, location permissions are enabled only when necessary and handled as precisely as possible. Most sensitive data involves emails or other personally identifiable information, and we take care to manage it in compliance with privacy regulations. (P7)"*

> *"We use encryption in transit, like TLS and HTTPS, standard contractual clauses, and data deletion policies. Once somebody gives us their information, it's only collected in the database for the individual company and not stored on our side, except perhaps minimally on the back end. (P1)"*

> *"My checklist includes encryption, minimal data collection, regular updates, developmental control, and documentation of any potential risk that you see. (P10)"*

Using automated compliance tools. They also leverage tools for automated compliance checks:

> *"We use CodeQL to scan every SDK, and GitGuardian to check that no secrets are pushed to GitHub. If there are no issues with CodeQL, we can commit the SDK to a repository and use it. (P5)"*

Whitelists are often integrated into broader management tools such as a Tech Catalog, which lists recommended, conditionally allowed, and prohibited SDKs:

> *"We have something called the Tech Catalog at my company. It lists all the libraries and SDKs you can potentially use based on the solution type. It shows the most recommended SDKs, some that can be used if necessary, and others that are absolutely prohibited. Before using an SDK, we submit a solution review, and the team ensures nothing is missing before we are allowed to use it. (P3)"*

4.2.5 SDK Transparency Practices. In the survey, SDK developers predominantly adhere to platform-specific privacy disclosure requirements, with a strong emphasis on iOS compliance. The vast majority reported providing iOS Privacy Manifests and following iOS Privacy Label Disclosure Guidelines, each selected by 125 participants (99.2%), reflecting the possibility of multiple selections per participant. Privacy Policies were provided by 96 participants (76.2%), and Terms of Service by 89 participants (70.6%). On Android, 63 participants (50.0%) followed the Data Safety Section Disclosure Guidelines, while 38 participants (30.2%) provided a Software Bill of Materials (SBOM). Very few participants indicated that their SDKs provided no privacy disclosures or selected "Other" (2 participants each, 1.6%).

Furthermore, interview participants highlighted additional transparency practices.

Continuous risk assessments for SDK privacy behaviors. Some participants encouraged to monitor SDK behavior and data collection, ensuring that any deviations from privacy standards are addressed. This can involve reviewing network calls, consulting documentation, or contacting SDK providers to implement safer alternatives:

> *"I have come across SDKs that collect more data than expected or lack clear documentation. In such cases, I pass the integration, review network calls and documentation, and sometimes contact the SDK provider. If the data is unnecessary or violates privacy standards, I implement safeguards or choose a safer alternative. (P2)"*

Operationalizing SDK data transparency and disclosure. SDK developers emphasized that transparency is a critical aspect of privacy compliance. They described how disclosures are embedded in applications and communicated to clients:

> *"We have disclosures when we build the app for our potential customers. When people input their information, it is clear that the data is collected strictly for the management company's purposes. It is not sold or sent to other parties, and we follow security procedures. This information is communicated to our clients, who then communicate it to their potential renters. (P3)"*

Transparency is also reinforced through the use of approved SDKs from reputable providers:

> *"It's up front. We use the approved list because they are transparent and upfront. It is never too vague to where we don't understand how our information is being handled, processed, or stored. (P3)"*

> *"Most big, reputable SDK providers are very clear. Their licensing, documentation, and data collection practices are accurate and straightforward. We avoid beta SDKs or less reputable SDKs. (P4)"*

Store-enforced transparency practices. They also noted that privacy requirements from app stores enforce transparency:

> *"If you are going to publish an app to the App Store or Google Play Store, you must detail exactly how you will use a user's data in your application. This ensures transparency and reasonable disclosure. (P2)"*

Some developers emphasized internal practices for SDK design, such as careful access specifiers and permission handling:

> *"When developing an SDK, access specifiers matter. Defining functions as public, internal, or private ensures that data cannot be manipulated in unintended ways. This is an important consideration for transparency and privacy. (P5)"*

Based on the quantitative survey data, we asked if there is any association between SDK types and perceived compliance effectiveness.

4.2.6 Association Between SDK Types and Compliance Practices. We investigated how SDK types and functionalities influence developers' reported support mechanisms, the ease and effectiveness of integrating their SDKs into applications, and the associated integration challenges. The analysis shows that certain SDK types and functionalities are significantly associated with the availability of developer support resources and integration burden. For instance, Web SDKs were positively associated with stronger development support resources (OR = 2.58, 95% CI: 1.14–5.86, $p = 0.023$), while Analytics SDKs also predicted greater availability of documentation resources (OR = 2.28, 95% CI: 1.03–5.05, $p = 0.043$). Cross-platform SDKs increased the likelihood of providing educational resources (OR = 2.58, 95% CI: 1.12–5.91, $p = 0.025$), and Developer-focused SDKs were linked to stronger community support mechanisms (OR = 2.40, 95% CI: 1.02–5.60, $p = 0.043$). Payments SDKs were associated with a lower likelihood of excessive manual work (OR = 0.37, 95% CI: 0.16–0.86, $p = 0.021$). For integration difficulty, extremely

difficult integration was less likely (OR = 0.045, 95% CI: 0.014–0.148, $p < 0.001$), while slightly difficult integration was more likely (OR = 3.78, 95% CI: 1.36–10.5, $p = 0.011$). Table 6 presents the statistically significant associations between SDK types and integration-related outcomes.

Table 3: Ordinal Logistic Regression Results for Perceived Compliance Practice Effectiveness

Predictor	Odds Ratio	95% CI	p-value
SDK Platforms			
Web	**2.31**	**[1.10, 4.87]**	**0.025**
Android	*1.40*	*[0.70, 2.80]*	*0.388*
iOS	*2.07*	*[0.99, 4.35]*	*0.060*
Cross-platform	1.49	[0.64, 3.48]	0.285
Nintendo Switch	13.57	[0.53, 348.7]	0.149
Primary SDK Types			
Analytics/Tracking	1.22	[0.63, 2.37]	0.579
Advertising/Monetization	1.33	[0.66, 2.67]	0.485
Social/Sharing Features	*1.90*	*[0.93, 3.87]*	*0.099*
Authentication/Security	1.46	[0.68, 3.13]	0.294
Developer Utilities	*0.54*	*[0.27, 1.09]*	*0.091*
Payments/Finance	1.54	[0.74, 3.20]	0.262
Other SDKs	*4.87*	*[0.90, 26.30]*	*0.102*

4.3 RQ3: Challenges in SDK Privacy Compliance

4.3.1 Data transparency. The survey results indicate that SDK developers encounter a range of challenges when attempting to achieve privacy compliance, spanning regulatory complexity, uncertainty introduced by third-party components, limited tooling, and pressure from product timelines. Specifically, the majority of survey participants reported difficulty keeping up with regulatory compliance updates, such as GDPR, with 66.9% selecting this option. Over half of the participants noted challenges related to assessing third-party behavior (52.1%), while 47.1% reported encountering a lack of clarity or examples and the burden of excessive manual work due to poor tooling. Only a small fraction indicated that they had not experienced significant challenges (5.8%) or selected other challenges (1.7%).

In the interviews, beyond legal compliance adoption, developers also expressed frustration with the transparency practices of third-party SDKs they are required to rely on.

Lack of opacity in third-party SDK data practices. Developers reported facing challenges when SDKs are not fully transparent about data collection. One participant described:

> *"Sometimes SDKs collect more data than documented, such as device identifiers or location data, without clear consent. They can be vague about their practices, and it may be difficult to properly handle opt-outs or data deletion requests. This makes it challenging for developers like myself to comply with privacy regulations, especially when dependencies increase app size or slow performance. (P1)"*

Developers highlighted several risks associated with using third-party SDKs, particularly those from less formal sources such as GitHub:

> *"While some SDKs may appear suitable, it is difficult to know exactly what is happening inside the source code. I try to use SDKs from authentic, officially linked sources, such as verified developer portals, to reduce the risk of extra data collection or misuse of permissions. GitHub projects, while auditable and supported by a community for code review, still require careful scrutiny before integration. (P3)"*

Reliance on internal privacy mechanisms. Also, they emphasized that they are mostly relying on their transparency mechanism:

> *"It depends on the vendor. Some are very transparent and provide detailed documentation about what data is collected, stored, and used. Others use vague language, such as "we may collect some data" without specifying what or how. I feel that these practices are somewhat deceptive, which prompts me to investigate further before integrating the SDK. (P9)"*

Interestingly, developers also emphasized that while SDK vendors may be transparent, end users often do not read the documentation fully, which makes it challenging for developers to implement data transparency practices that are compliant with applicable laws. For example, participants said:

> *"Compliance means that any data collected from a user must be for the purpose declared in the privacy policy. Data should have a retention policy, such as deletion or anonymization six months after customer offboarding. (P7)"*

> *"SDKs are generally fully transparent according to what they need to declare, but many users do not read the documentation thoroughly. Legal language can make it hard to understand. (P6)"*

4.3.2 Balance between product delivery timelines and compliance. According to the survey results, balancing privacy compliance with product delivery timelines is perceived as a moderately demanding aspect of SDK development for most respondents. The majority of respondents found this task moderately challenging, with 54 participants (42.9%) selecting moderately challenging. A substantial portion rated it as slightly challenging (32; 25.4%) or very challenging (22; 17.5%). Fewer participants indicated that it was not challenging (12; 9.5%) or extremely challenging (5; 4.0%), while 1 participant (0.8%) selected not applicable.

The interviews not only supported the survey findings but also revealed additional challenges in product development, including reliance on outdated protocols over time.

Challenges in lack of regulatory compliance guidelines. SDK developers noted that technical and regulatory issues often contribute to the difficulty of balancing compliance with product delivery. Compatibility between platforms, such as iOS and Android, and reliance on older security protocols were frequently cited as complicating factors:

> *"Um, I mean, mainly compatibility issues when it comes to, like, the platforms that are utilized, um, like between iOS and Android and stuff like that. Just compatibility issues, outdated TLS, um… weak encryption at times. (P1)"*

Challenges in implementing technical safeguards. In addition, developers highlighted challenges related to permission settings and the implementation of encryption. Weak or improperly configured code can create delays, as developers need to ensure that privacy standards are met without affecting product delivery:

> *"Sometimes permission settings, if that makes sense? Um, like, the permissions… the… of the type of data that's being collected and encrypted. Yeah, I'm talking about, like, errors in… weak source code that undermines the encryption process that does not meet the standards that my company wants to utilize. (P1)"*

4.3.3 Monitoring practices. Our survey study shows that although a substantial portion of SDK developers find third-party data monitoring moderately challenging, only a few consider it trivial or extremely difficult, highlighting the nuanced nature of balancing oversight with development constraints. Specifically, the majority of respondents indicated that this task is somewhat difficult, with 46.0% reporting "Somewhat difficult." Approximately 27.8% of participants selected "Neither easy nor difficult," while 17.5% found it "Somewhat easy." A smaller proportion of respondents reported extreme difficulty, with 6.3% indicating "Extremely difficult" and only 2.4% selecting "Extremely easy."

4.3.4 Implementing compliance features. Survey participants were asked to rate how challenging it is to implement privacy compliance features in SDK development, such as consent management and data minimization. The majority of participants found implementation to be somewhat challenging. Specifically, 58 participants (46%) reported it as moderately challenging, followed by 37 participants (29.4%) who found it slightly challenging, and 15 participants (11.9%) who reported it as very challenging. Fewer participants indicated it was not challenging (11; 8.7%) or extremely challenging (4; 3.2%), and 1 participant (0.8%) selected not applicable.

One of the most common themes emerging from the interviews was the limited availability of compliance tools for implementation.

Limitations of available compliance tools. Developers also noted the limitations of SDK-level compliance features and the importance of following platform guidelines:

> *"I think it's not possible, specifically when it comes to the iOS world, because Apple has actually pointed out that some SDKs do not have clear policies. I think just sticking with what Apple and Google require is more than sufficient. For example, if you can use any SDK and push it to Apple, there might be an issue—Apple is already validating your app when you upload it to the App Store. (P8)"*

Based on the quantitative survey data, we further investigated to understand the association of SDK developers' team size and privacy compliance practices.

4.3.5 Association Between Team Size and Compliance Practices. We examined the association between team size and the adoption of various privacy compliance practices using logistic regression. Odds ratios were estimated for each practice, comparing different team size levels with the smallest teams as the reference category. For the majority of practices, odds ratios were not statistically significant ($p > 0.05$), suggesting that team size alone does not reliably predict adoption. Some practices, however, showed significant associations with team size. Smaller teams were significantly less likely to implement internal privacy reviews (OR = 0.236, 95% CI: 0.080–0.698, $p = 0.009$) and maintain formal privacy policies (OR = 0.124, 95% CI: 0.034–0.454, $p = 0.0016$). Larger teams (20+ members) were more likely to adopt platform compliance measures such as iOS privacy manifests (SDK transparency) (OR = 3.29, 95% CI: 1.34–8.04, $p = 0.0092$) and iOS privacy label disclosures (SDK transparency) (OR = 2.73, 95% CI: 1.13–6.58, $p = 0.0253$). Other practices related to SDK development lifecycle activities and related compliance practices showed variable odds ratios across team sizes but did not reach statistical significance (ORs ranged from 0.837 to 6.18, all $p > 0.05$). Table 4 presents the statistically significant associations between team size and compliance practices.

Table 4: Significant associations between team size and privacy-related practices among SDK developers

Practice	Privacy	Team	OR	95% CI	p
Internal Governance	Internal Review	11-20	**0.24**	[0.08, 0.70]	0.009
Platform Compliance	Apple/Google Play	Independent	**0.16**	[0.03, 0.93]	0.042
SDK Transp.	SBOM	20+	**6.67**	[1.38, 32.2]	0.018
SDK Transp.	iOS Label	6-10	**3.29**	[1.34, 8.04]	0.009
Organizational Transp.	Privacy Policies	11-20	**0.12**	[0.03, 0.45]	0.002
Organizational Transp.	Privacy Policies	Independent	**0.15**	[0.02, 0.94]	0.043
Organizational Transp.	Terms of Service	11-20	**0.17**	[0.05, 0.54]	0.003
Organizational Transp.	Terms of Service	20+	**0.15**	[0.03, 0.70]	0.017
	iOS Label	6-10	**2.73**	[1.13, 6.58]	0.025

4.3.6 Association Between Team Size and Compliance Challenges. We examined the association between SDK developers' team size and compliance challenges related to privacy requirements. Several team size levels showed significant associations with balancing product delivery timelines and compliance requirements. Specifically, respondents from medium-sized teams (6–10 people) had higher odds of reporting implementation challenges ($OR = 4.49, p = 0.011$), whereas respondents from very large teams (20+ people) had lower odds ($OR = 0.099, p < 0.001$). Similarly, assessing the ease or difficulty of SDK integration showed significant effects of team size. Small teams (1–5 people) were less likely to perceive integration as difficult ($OR = 0.332, p = 0.025$), whereas very large teams (20+ people) reported substantially higher odds of perceiving difficulty ($OR = 7.60, p < 0.001$). Finally, for implementing compliance support features, community support mechanisms was significantly associated with medium-sized teams (6–10 people) ($OR = 0.345, p = 0.041$). Other support options, such as documentation, SDK-specific templates, and automated tools, did not show significant differences by team size, and none of the SDK transparency-related challenges were significantly associated with team size ($p > 0.05$ for all options). Table 5 presents the statistically significant associations between team size and compliance-related difficulties.

Table 5: Significant associations between team size and SDK development practices. Odds ratios (OR), 95% confidence intervals (CI), and p-values are shown.

Practice	Team Size	OR	95% CI	p
Implementation	6-10	4.49	[1.40, 14.4]	0.011
Implementation	20+	0.099	[0.027, 0.365]	<0.001
Integration difficulty	1-5	0.332	[0.127, 0.872]	0.025
Integration difficulty	20+	7.60	[4.29, 13.5]	<0.001
Community support	6-10	0.345	[0.125, 0.955]	0.041

4.3.7 Association Between SDK Types and Compliance Practices. We investigated the influence of SDK types and functionalities on developers' reported compliance-related practices and integration challenges. The analysis shows that certain SDK types and functionalities are significantly associated with its development support and integration . For instance, Web SDKs were positively associated with better documentation and API support (stronger development support resources) (OR = 2.58, 95% CI: 1.14–5.86, $p = 0.023$), while Analytics SDKs also predicted improved documentation (availability of documentation resources) (OR = 2.28, 95% CI: 1.03–5.05, $p = 0.043$). Cross-platform SDKs increased the likelihood of providing educational resources (OR = 2.58, 95% CI: 1.12–5.91, $p = 0.025$), and Developer-focused SDKs were linked to stronger community support mechanisms (OR = 2.40, 95% CI: 1.02–5.60, $p = 0.043$). Payments SDKs were associated with a lower likelihood of excessive manual work (OR = 0.37, 95% CI: 0.16–0.86, $p = 0.021$). For integration difficulty, extremely difficult integration was less likely (OR = 0.045, 95% CI: 0.014–0.148, $p < 0.001$), while slightly difficult integration was more likely (OR = 3.78, 95% CI: 1.36–10.5, $p = 0.011$). Table 6 presents the statistically significant associations between SDK types and integration-related outcomes.

Table 6: Significant associations between SDK types/functionality and compliance practices

Practice	SDK Type	OR	CI	p
API Documents	Web	2.58	[1.14, 5.86]	0.023
API Documents	Analytics	2.28	[1.03, 5.05]	0.043
Tutorials	Cross-platform	2.58	[1.12, 5.91]	0.025
Community	Dev-focused	2.40	[1.02, 5.60]	0.043
Less Difficult Manual Integration	Payments	0.37	[0.16, 0.86]	0.021
Extremely Difficult Integration ↓	—	0.045	[0.014, 0.148]	<0.001
Slightly Difficult Integration↑	—	3.78	[1.36, 10.5]	0.011

5 Recommendations

Based on our findings, we provide actionable recommendations organized according to our three research questions.

Addressing SDK developers' responsibilities in privacy compliance. While 78.6% of SDK developers believe they should be responsible for privacy compliance, our interviews revealed that developers primarily follow standards set by legal and security teams rather than determining compliance themselves. To address this disconnect, organizations should formally document the division of privacy compliance responsibilities between SDK developers, legal teams, and security teams, specifying who is responsible for interpreting requirements, implementing controls, and maintaining documentation. Organizations should establish regular communication mechanisms to ensure SDK developers receive timely guidance on evolving regulatory requirements. Regulators should provide clear guidance on how compliance responsibility is distributed when multiple parties (SDK developers, app developers, platform providers) are involved in data processing, including joint controller relationships and liability allocation.

Improving privacy-compliant SDK development practices. SDK developers incorporate privacy compliance throughout the development lifecycle, most frequently using compliance frameworks (77.7%) and internal privacy reviews (71.9%). However, significant gaps exist in organizational privacy governance, platform compliance practices, and SDK transparency measures, particularly across different team sizes. Organizations should mandate privacy impact assessments during the design phase for all SDKs, particularly those handling sensitive data categories. They should integrate tools such as CodeQL and GitGuardian into CI/CD pipelines to detect potential privacy violations before deployment. Smaller teams (1–10 developers) were significantly less likely to implement internal privacy reviews (OR = 0.236, $p = 0.009$) and maintain formal privacy policies (OR = 0.124, $p = 0.0016$), while larger teams (20+) were more likely to adopt SDK transparency measures such as SBOM documentation and iOS privacy label disclosures. Organizations should provide lightweight compliance checklists and automated tooling for smaller teams, while larger teams should invest in dedicated privacy engineering expertise. The developer community needs tools specifically designed for SDK contexts, including automated privacy manifest generators, cross-library data flow analyzers, and platform-agnostic compliance frameworks.

Mitigating integration challenges and supporting diverse team structures. SDK developers face substantial challenges, with 66.9% reporting difficulty keeping up with regulatory updates, 52.1% struggling to assess third-party behavior, and 85.8% finding it challenging to balance compliance with delivery timelines. Team size significantly influences these challenges, with medium teams facing higher implementation difficulties (OR = 4.49, p = 0.011) and large teams facing greater integration difficulties (OR = 7.60, p < 0.001). Organizations should assign responsibility for monitoring privacy law changes and translating them into actionable technical requirements through regulatory update services or privacy counsel. Organizations should deploy continuous monitoring solutions that track network traffic, analyze permission requests, and compare actual SDK behavior against documented privacy policies. Privacy compliance activities should be embedded into sprint planning and definition of done criteria rather than treating compliance as a separate, end-stage gate. Privacy requirements should be included in acceptance criteria and privacy reviews should be conducted as part of code review processes. Platform providers should enhance development tools with real-time privacy feedback through IDE plugins, privacy linting tools, and automated privacy testing during app submission.

6 Related Work

Qualitative studies on privacy compliance practices. In recent years, the literature has extensively examined privacy compliance practices in software engineering, mainly from the perspective of mobile application developers. Mixed-methods studies of mobile app developers report reliance on checklists, policy templates, and post-hoc fixes, frequently driven by external enforcement pressure rather than proactive design [6]. Studies focused on child-directed applications demonstrate that developers face substantial challenges in identifying personal data flows, managing third-party services, and implementing legally mandated safeguards under regulations such as COPPA [7]. Similarly, GDPR-focused qualitative studies identify difficulties in implementing requirements related to data minimization, consent, deletion, and purpose limitation, particularly under time and resource constraints [20]. Li et al. [28] investigate the challenges app developers face when creating accurate privacy nutrition labels, as required by modern app marketplaces. Through interviews and empirical analysis, they find that developers struggle with understanding data flows, interpreting legal requirements, and mapping SDK behaviors to disclosure categories, resulting in frequent inaccuracies even when developers intend to comply. Samarin et al. [41] assess Android app developers' compliance with the CCPA, focusing on both privacy notices and responses to "right to know" requests. Tahaei and Vaniea [45] analyze the privacy guidance provided by major advertising networks to Android developers through a walkthrough study with an experienced developer.

Compared to prior work, we conducted a human-subject study with a less-explored participant group, i.e., SDK developers, to gain a deeper understanding of privacy non-compliance from the perspective of how mobile software dependencies shape these practices.

Privacy analysis on mobile SDKs. A large body of measurement-focused literature shows that third-party SDKs and libraries play a significant role in shaping applications' privacy behaviors [8, 13, 17, 25, 26, 29, 30, 42, 43, 47, 54]. To measure the privacy leakage caused by the widespread integration of SDKs, prior research has utilized both static [34] and dynamic [15, 37, 38] program analysis, revealing that sensitive user data collection is prevalent. Derr et. al [17] have identified that outdated library versions often contribute to privacy risks, and several auto-update techniques of SDKs have been proposed to address this issue [18, 21, 24]. Xiao et.al [50] systematically investigate privacy compliance issues of SDK's privacy label disclosure guideline and found the prevalence of privacy disclosure non-compliance issues. In addition, [7, 32, 39, 40, 44, 52, 53] have discovered that misconfigurations of SDKs by app developers can result in the leakage of sensitive personal information.

Different from these work, our study aims at understanding SDK developers' perspective, procedures and strategies for privacy compliance assessment within the mobile software supply chain.

7 Conclusion

In this study, we investigated privacy awareness and practices among mobile SDK developers through a mixed-methods approach combining a survey of 126 SDK developers and semi-structured interviews with 10 practitioners.

SDK developers broadly recognize their privacy compliance obligations, yet face a persistent gap between perceived responsibility and actual authority, with compliance decisions largely delegated to legal and security teams (**RQ1**). Privacy practices are incorporated across the development lifecycle but are constrained by team size, regulatory complexity, and third-party opacity (**RQ2**, **RQ3**).

In addition, we discussed the challenges involved in implementing privacy-compliant SDK. We also provided recommendations for clearer accountability, better tools and automation, and support for diverse team structures to maintain user privacy while enabling sustainable SDK development.

Ethical Considerations

This study was approved by our institute's Institutional Review Board (IRB) and posed minimal risks to the study participants. Prior to the survey, we ensured that participants were informed about the purpose of the study and the consent process. They were acknowledged that the study was entirely voluntary and had the right to opt-out and leave the survey and interview at any time. Additionally, participants could skip any questions they did not feel comfortable answering. Participants were also informed that the data would be used solely for research purposes and deleted in accordance with the data retention guidelines described in the Informed Consent Statement. All personally identifiable information (PII) collected during the study was removed to protect participants' confidentiality and comply with ethical guidelines for research involving human subjects. No participants raised concerns regarding compensation or participation risks in their final free-text responses, nor did they contact the research team or the IRB for feedback.

Open Science

In accordance with the conference's open science policy, we make our study artifacts available to support transparency and reproducibility. Specifically, we release (1) the recruitment materials, (2) the full survey questionnaire, (3) the semi-structured interview guide, and (4) the qualitative codebook as supplementary material, archived at https://doi.org/10.5281/zenodo.21465694.

Generative AI Usage

The authors acknowledge the use of AI-based tools, specifically GPT-5 and Gemini, for grammar refinement, language editing, and improving the clarity and readability of the text in this manuscript. All intellectual contributions, content development, and research findings presented in this paper remain the sole responsibility of the authors.

References

[1] Children's online privacy protection act (coppa), 15 u.s.c. §§ 6501–6506, 1998. United States Federal Law.
[2] General data protection regulation (gdpr), regulation (eu) 2016/679, 2016. Official Journal of the European Union.
[3] California consumer privacy act (ccpa), cal. civ. code § 1798.100 et seq., 2018. California State Legislature.
[4] Personal information protection law (pipl) of the people's republic of china, 2021. Adopted by the Standing Committee of the National People's Congress.
[5] Greenley v. kochava inc. No. 22-CV-01327-BAS-AHG, 2023 WL 4833466, at *1 (S.D. Cal.July 27, 2023), 2023.

[6] Abdulrahman Alhazmi and Nalin Asanka Gamagedara Arachchilage. I'm all ears! listening to software developers on putting gdpr principles into software development practice. *Personal and Ubiquitous Computing*, 25(5):879–892, 2021.

[7] Noura Alomar and Serge Egelman. Developers say the darnedest things: Privacy compliance processes followed by developers of child-directed apps. *Proceedings on Privacy Enhancing Technologies*, 2022.

[8] Michael Backes, Sven Bugiel, and Erik Derr. Reliable third-party library detection in android and its security applications. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 356–367, 2016.

[9] Reuben Binns, Ulrik Lyngs, Max Van Kleek, Jun Zhao, Timothy Libert, and Nigel Shadbolt. Third party tracking in the mobile ecosystem. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '18, page 23–31, New York, NY, USA, 2018. Association for Computing Machinery.

[10] Theodore Book, Adam Pridgen, and Dan S Wallach. Longitudinal analysis of android ad library permissions. *arXiv preprint arXiv:1303.0857*, 2013.

[11] Virginia Braun and Victoria Clarke. *Thematic analysis*, page 57–71. APA handbooks in psychology®. American Psychological Association, Washington, DC, US, 2012.

[12] Ann Cavoukian et al. Privacy by design: The 7 foundational principles. *Information and privacy commissioner of Ontario, Canada*, 5(2009):12, 2009.

[13] Yue Chen, Yulong Zhang, Zhi Wang, Liangzhao Xia, Chenfu Bao, and Tao Wei. Adaptive android kernel live patching. In *USENIX Security Symposium*, pages 1253–1270, 2017.

[14] Juliet M Corbin and Anselm Strauss. Grounded theory research: Procedures, canons, and evaluative criteria. *Qualitative sociology*, 13(1):3–21, 1990.

[15] Jonathan Crussell, Ryan Stevens, et al. Madfraud: Investigating ad fraud in android applications. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 123–134, 2014.

[16] George Danezis, Josep Domingo-Ferrer, Marit Hansen, Jaap-Henk Hoepman, Daniel Le Metayer, Rodica Tirtea, and Stefan Schiffner. Privacy and data protection by design-from policy to engineering. *arXiv preprint arXiv:1501.03726*, 2015.

[17] Erik Derr, Sven Bugiel, Sascha Fahl, Yasemin Acar, and Michael Backes. 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*, pages 2187–2200, 2017.

[18] Yue Duan, Lian Gao, Jie Hu, and Heng Yin. Automatic generation of non-intrusive updates for third-party libraries in android applications. In *RAID*, pages 277–292, 2019.

[19] European Commission. Communication from the commission to the european parliament and the council: Data protection rules as a trust-enabler in the EU and beyond – taking stock. COM(2019) 374 final, July 2019. Brussels, 24.7.2019.

[20] Lucas James Franke. *Understanding the Impact of Data Privacy Regulations on Software and Its Stakeholders*. PhD thesis, Virginia Tech, 2023.

[21] Jie Huang, Nataniel Borges, Sven Bugiel, and Michael Backes. Up-to-crash: Evaluating third-party library updatability on android. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 15–30. IEEE, 2019.

[22] Simon Koch, Manuel Karl, Robin Kirchner, Malte Wessels, Anne Paschke, and Martin Johns. The impact of default mobile sdk usage on privacy and data protection. *Proceedings on Privacy Enhancing Technologies*, 2025.

[23] Simon Koch, Malte Wessels, Benjamin Altpeter, Madita Olvermann, and Martin Johns. Keeping privacy labels honest. *Proceedings on Privacy Enhancing Technologies*, 2022.

[24] Bodong Li, Yuanyuan Zhang, Juanru Li, Runhan Feng, and Dawu Gu. Appcommune: Automated third-party libraries de-duplicating and updating for android apps. In *2019 IEEE 26th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 344–354. IEEE, 2019.

[25] Li Li, Tegawendé F Bissyandé, Jacques Klein, and Yves Le Traon. An investigation into the use of common libraries in android apps. In *2016 IEEE 23Rd international conference on software analysis, evolution, and reengineering (SANER)*, volume 1, pages 403–414. IEEE, 2016.

[26] Menghao Li, Wei Wang, Pei Wang, Shuai Wang, Dinghao Wu, Jian Liu, Rui Xue, and Wei Huo. Libd: Scalable and precise third-party library detection in android markets. In *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, pages 335–346. IEEE, 2017.

[27] Tianshi Li, Elizabeth Louie, Laura Dabbish, and Jason I. Hong. 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):220:1–220:28, January 2021.

[28] Tianshi Li, Kayla Reiman, Yuvraj Agarwal, Lorrie Faith Cranor, and Jason I Hong. Understanding challenges for developers to create accurate privacy nutrition labels. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, pages 1–24, 2022.

[29] Bin Liu, Bin Liu, Hongxia Jin, and Ramesh Govindan. Efficient privilege deescalation for ad libraries in mobile apps. In *Proceedings of the 13th annual international conference on mobile systems, applications, and services*, pages 89–103, 2015.

[30] Ziang Ma, Haoyu Wang, Yao Guo, and Xiangqun Chen. Libradar: Fast and accurate detection of third-party libraries in android apps. In *Proceedings of the 38th international conference on software engineering companion*, pages 653–656, 2016.

[31] Mark Huasong Meng, Chuan Yan, Yun Hao, Qing Zhang, Zeyu Wang, Kailong Wang, Sin Gee Teo, Guangdong Bai, and Jin Song Dong. A large-scale privacy assessment of android third-party sdks. *arXiv preprint arXiv:2409.10411*, 2024.

[32] Abraham H Mhaidli, Yixin Zou, and Florian Schaub. " 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)*, pages 225–244, 2019.

[33] Matthew Miles, Michael Huberman, and Johnny Saldana. *Qualitative Data Analysis: A Methods Sourcebook*. SAGE Publications, Inc., 4th edition, 2014.

[34] Yuhong Nan, Zhemin Yang, Xiaofeng Wang, Yuan Zhang, Donglai Zhu, and Min Yang. Finding clues for your secrets: Semantics-driven, learning-based privacy discovery in mobile apps. In *NDSS*, 2018.

[35] Joanne Neale. Iterative categorization (ic): a systematic technique for analysing qualitative data. *Addiction*, 111(6):1096–1106, 2016.

[36] Stuart L Pardau. The california consumer privacy act: Towards a european-style privacy regime in the united states. *J. Tech. L. & Pol'y*, 23:68, 2018.

[37] Abbas Razaghpanah, Rishab Nithyanand, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Mark Allman, Christian Kreibich, Phillipa Gill, et al. 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)*, 2018.

[38] Jingjing Ren, Martina Lindorfer, Daniel J Dubois, Ashwin Rao, David Choffnes, and Narseo Vallina-Rodriguez. A longitudinal study of pii leaks across android app versions. In *Proceedings of the 25th Network and Distributed System Security Symposium (NDSS 2018)*, 2018.

[39] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, and Serge Egelman. "won't somebody think of the children?" examining coppa compliance at scale. July 2018. Accepted: 2021-07-13T09:33:32Z.

[40] David Rodriguez, Joseph A Calandrino, Jose M Del Alamo, and Norman Sadeh. Privacy settings of third-party libraries in android apps: A study of facebook sdks. *Proceedings on Privacy Enhancing Technologies*, 2025.

[41] Nikita Samarin, Shayna Kothari, Zaina Siyed, Oscar Bjorkman, Reena Yuan, Primal Wijesekera, Noura Alomar, Jordan Fischer, Chris Hoofnagle, and Serge Egelman. Lessons in vcr repair: Compliance of android app developers with the california consumer privacy act (ccpa). (arXiv:2304.00944), April 2023. arXiv:2304.00944 [cs].

[42] Jaebaek Seo, Daehyeok Kim, Donghyun Cho, Insik Shin, and Taesoo Kim. Flexdroid: Enforcing in-app privilege separation in android. In *NDSS*, 2016.

[43] Shashi Shekhar, Michael Dietz, and Dan S Wallach. Adsplit: Separating smartphone advertising from applications. In *USENIX security symposium*, volume 2012, 2012.

[44] Mohammad Tahaei, Kopo M Ramokapane, Tianshi Li, Jason I Hong, and Awais Rashid. Charting app developers' journey through privacy regulation features in ad networks. *Proceedings on Privacy Enhancing Technologies*, 1:24, 2022.

[45] Mohammad Tahaei and Kami Vaniea. "developers are responsible": What ad networks tell developers about privacy. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI EA '21, page 1–11, New York, NY, USA, May 2021. Association for Computing Machinery.

[46] Paul Voigt and Axel Von dem Bussche. *The EU General Data Protection Regulation (GDPR): A Practical Guide*. Springer, 1st edition, 2017.

[47] Haoyu Wang and Yao Guo. Understanding third-party libraries in mobile app analysis. In *2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C)*, pages 515–516. IEEE, 2017.

[48] Jice Wang, Yue Xiao, Xueqiang Wang, Yuhong Nan, Luyi Xing, Xiaojing Liao, JinWei Dong, Nicolas Serrano, Haoran Lu, XiaoFeng Wang, et al. Understanding malicious cross-library data harvesting on android. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 4133–4150, 2021.

[49] Yue Xiao, Zhengyi Li, Yue Qin, Xiaolong Bai, Jiale Guan, Xiaojing Liao, and Luyi Xing. Lalaine: Measuring and characterizing Non-Compliance of apple privacy labels. page 1091–1108, 2023.

[50] Yue Xiao, Chaoqi Zhang, Yue Qin, Fares Fahad S Alharbi, Luyi Xing, and Xiaojing Liao. Measuring compliance implications of third-party libraries' privacy label disclosure guidelines. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pages 1641–1655, 2024.

[51] Shishuai Yang, Qinsheng Hou, Shuang Li, and Wenrui Diao. Do app developers follow the android official data security guidelines? an empirical measurement on app data security. In *2023 30th Asia-Pacific Software Engineering Conference (APSEC)*, page 71–80, December 2023.

[52] Xueling Zhang, Xiaoyin Wang, Rocky Slavin, Travis Breaux, and Jianwei Niu. How does misconfiguration of analytic services compromise mobile privacy? In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, pages 1572–1583, 2020.

[53] Yifan Zhang, Zhaojie Hu, Xueqiang Wang, Yuhui Hong, Yuhong Nan, XiaoFeng Wang, Jiatao Cheng, and Luyi Xing. Navigating the privacy compliance maze: Understanding risks with {Privacy-Configurable} mobile {SDKs}. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 6543–6560, 2024.

[54] Yuan Zhang, Jiarun Dai, Xiaohan Zhang, Sirong Huang, Zhemin Yang, Min Yang, and Hao Chen. Detecting third-party libraries in android applications with high precision and recall. In *2018 IEEE 25th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 141–152. IEEE, 2018.