Analysing Privacy Risks in Children's Educational Apps in Australia

Sicheng Jin, Rahat Masood, Jung-Sook Lee, and Hye-Young (Helen) Paik
University of New South Wales
Email: {stefan_zalkoszin.jin, rahat.masood, js.lee, h.paik}@unsw.edu.au

***Abstract*—The integration of educational technology (edtech) into primary and secondary schools has substantially accelerated, making digital applications core components of modern learning environments. While ostensibly beneficial, these apps introduce substantial privacy and security risks for children, frequently through opaque data collection and sharing practices. However, existing research on children's applications has predominantly relied on automated dynamic analysis tools which fail to replicate authentic human behaviours, such as navigating parental gates, configuring privacy settings, or specifically claiming as student or teacher. Furthermore, prior studies have largely overlooked the accessibility of privacy policies for non-legal experts and do not reflect the current practices of Australian education departments. This paper presents a comprehensive analysis of approximately 200 Android applications sourced from both Australian school recommendations and the Google Play Store's "Kids" and "Educational" categories. Our methodology follows three-stepped approach: (1) static analysis of application code; (2) dynamic analysis of live network traffic to observe real-world data transmissions; and (3) textual analysis of privacy policies to assess their readability and compare their disclosures against observed behaviour. The findings indicate that a substantial subset, 46% of apps, still engage in risky data practices, such as transmitting persistent identifiers not explicitly mentioned in their privacy policies. Additionally, these policies are typically written at a reading level above that of the average Australian parent. Our analysis shows that only 3% of privacy policies meet the threshold of being "fairly easy" to read, leaving most apps effectively inaccessible for parents. Policies rarely matched practice: only about 1 in 4 apps were fully consistent, while the remainder showed partial or conflicting disclosures, often omitting the information about third-party recipients and timing of collection. The vast majority (89.3%) of apps initiated outbound connections before any user activity on the apps. These findings offer crucial insights for educators, parents, developers, and policymakers in Australia and abroad to make informed decisions about selecting apps for children and shaping appropriate policy frameworks for educational apps.**

I. Introduction

Australian K–12 education is now deeply digital. The use of digital products for educational delivery and administrative services is ubiquitous in Australian schools [1]. In the Organisation for Economic Co-operation and Development (OECD) Programme for International Student Assessment (PISA) 2022, Australia is one of the few countries with at least one computer per student (compared to the OECD average of 0.81). Moreover, the South Australian department of education now mandates device-to-student ratios of 1:3 in primary and 1:1 in secondary by the end of 2026, institutionalising digital tools in everyday learning[1]. Similarly, in NSW and Victoria, system-level policies and school-level Bring Your Own Device (BYOD) programs normalise the use of third-party educational apps [2]. For example, NSW public schools commonly publish recommended iPad app lists for BYOD classes [3], [4], [5] and in Victoria, the department maintains a central software catalogue (Arc/eduSTAR) that schools draw on for classroom software [6], [7].

While these educational technologies offer advantages such as personalised learning and improved content delivery, they also participate in a data economy, in which the collection and commercialisation of user data is the main source of profit [1]. Many edtech companies engage in data collection practices that threaten the right to privacy of millions of children, collecting data well beyond what is necessary or appropriate, and in many cases, sharing this data with a murky list of third parties [8], [9]. This situation is worsened by the fact that the privacy policies, which are supposed to disclose these data collection practices, are often inconsistent to the actual behaviour of the app [10] and are written with such a high degree of legalism that they are impossible to understand for most users. Australian education is a regular breach target: in January–June 2024 the OAIC recorded 44 notifications from the Education sector, placing it among the top five most-affected industries [11]. A concrete case was the NSW Department of Education cyberattack in 2021, which forced systems offline ahead of Term 3 [12]. Risks also arise within the learning tools themselves: Human Rights Watch reported 145/163 (89%) government-endorsed EdTech products surveilled or had the capacity to surveil children [13], and a 2022 credential-stuffing incident on Seesaw allowed explicit links to be broadcast via school-home messaging [14]. These situations raise a major conundrum for schools and parents, which have a legal and moral obligation to protect the digital privacy of children.

Attributing these risks is a major challenge for educators and parents, who often lack the technical capability to interpret the black-boxed data practices of edtech vendors, resulting in

[1]https://www.education.sa.gov.au/

Symposium on Usable Security and Privacy (USEC) 2026
27 February 2026, San Diego, CA, USA
ISBN 978-1-970672-07-7
https://dx.doi.org/10.14722/usec.2026.23025
www.ndss-symposium.org, https://www.usablesecurity.net/USEC/

a "culture of compliance" rather than best-practice protection of students' rights [1]. Prior studies identified widespread potential COPPA violations, such as the sharing of persistent identifiers and poor use of third-party SDKs [8], [15]. While these studies have established valuable global benchmarks, several critical considerations have been largely overlooked by prior studies. Firstly, existing researches on child-focused applications mostly relied on automated dynamic analysis tools [15] which had distant usage patterns compared to a real human user. Secondly, prior works predominantly focused on the international extent, with less regard to the specifications of the Australian curricula. We also noticed that few works addressed the readability of policies, on whether they reflect the average literacy level of Australian parents.

there is a need for a focused and up-to-date analysis of the specific apps used by Australian children. These are the apps that schools actively adopt, operate within Australia's distinct legal and assurance frameworks (such as the Privacy Act and Safer Technologies 4 Schools (ST4S), and remain comparatively under-examined relative to those in US and EU contexts, meaning that insights from such an analysis would directly inform local procurement and child-safeguarding decisions.

In this paper, we make a contribution to the literature by closing these gaps. We conduct an analysis of around 200 Android apps that are relevant to Australian children. Our methodology involves three distinct analytical techniques to provide a holistic view of each app's behaviour: (i) *Static Analysis* of the application's source code to identify potential risks; (ii) *Differential Dynamic Analysis* of live network traffic to observe the app's actual data transmission behaviour in different states; and (iii) *Privacy Policy Analysis* to assess readability using quantitative metrics such as Flesch-Kincaid Grade Level and Gunning Fog Index, and to detect discrepancies between disclosed practices and observed behaviours. Through this comprehensive approach, we seek to answer the following research questions:

- RQ1: How do Australian-curriculum–focused apps currently handle user privacy in their design and operation, including their practices around data disclosure to third parties?
- RQ2: How prevalent are common security misconfigurations, such as active hardcoded secrets or insecure data transmission, in Australian-curriculum–focused apps?
- RQ3: To what extent do the privacy policies of Australian-curriculum–focused apps accurately reflect their actual data collection practices?
- RQ4: How accessible are the privacy policies of Australian-curriculum–focused app? and to what extent are the observed behaviours of these apps aligned with their privacy policies?

Our analysis reveals a pervasive ecosystem within Australian classrooms. We find that 89.3% of apps initiate idle telemetry transmitting data to third parties immediately upon first opening and before any user consent can be obtained. Furthermore, 83.6% of apps transmit persistent identifiers (such as Advertising IDs or Firebase IDs), often linking them across sessions in ways that potentially violate COPPA standards. Contrary to expectations, apps explicitly branded for children (e.g., "Kids," "Preschool") were not safer than general-audience tools; in fact, they were less likely to align with their privacy policies, with 76% of child-targeted apps exhibiting undisclosed or contradictory data practices. Finally, we find that informed consent is structurally impossible: 97% of privacy policies required university-level reading skills, causing them inaccessible to the average parent or teacher.

The remainder of this paper is structured as follows. Section II reviews the regulatory background and related literature work. Section III details our four-pronged data collection methodology. Section IV presents our findings for each research question. Section V discusses the implications of our findings, and Section VI concludes the paper.

II. Background and Related Works

This study sits between privacy regulation, technical analysis of app behaviours, and the socio-technical context of educational technology. We provide an overview of the legal and policy landscape around children's data, followed by a review of the academic literature on the technical analysis of app behaviours, the privacy challenges in education in particular, and the views of parents and developers.

A. Legal and Platform Policy Frameworks

Children's data online is protected by a mosaic of national laws and platform policies. Building on the pioneering work of the Children's Online Privacy Protection Act (COPPA), the former stipulates rigorous requirements for operators of online services directed at children under the age of 13. This includes clearly describing data practices and obtaining verifiable parental consent before collecting most forms of Personally Identifiable Information (PII) [8]. The PII term is defined widely to include not only contact information and geolocation, but also persistent identifiers such as cookies or device IDs if they are used to recognise a user over time and across services [9].

Recognising both the legal requirements and public pressure stipulated by these regulations, app marketplaces have developed their own policies. To participate in the "Designed for Families (DFF)" program, developers of child-directed apps must attest to their compliance with COPPA [15]. Perhaps more concerning, in compliance with COPPA, the DFF program restricts the collection of the Android Advertising ID (AAID) and precise location data from children and compels the use of only Google-certified ad Software Development Kits (SDKs) [15]. Additionally, the App Store Review Guidelines for apps in the Kids Category prohibit containing third-party advertising or analytics, and transmitting personally identifiable information to third parties [16].

The General Data Protection Regulation (GDPR) (Regulation (EU) 2016/679) is the European Union's comprehensive data-protection law. [17] It governs the processing of personal

data of individuals in the EU/EEA and can apply extraterritorially when organisations offer goods/services to, or monitor the behaviour of, people in the EU/EEA. It sets core principles (e.g., lawfulness, fairness, transparency, purpose limitation, data minimisation) and requires a lawful basis for processing. The GDPR also includes heightened protections for children (e.g., a "digital consent" age set by Member States between 13 and 16). We mention the GDPR here only to contextualise terminology commonly used in app privacy policies; our study does not rely on, or assess, GDPR compliance.

In Australia, the Privacy Act 1988 provides the overarching legal framework for data privacy. While not child-specific, its principles guide the handling of personal information. More specific guidance is provided by initiatives such as the Safer Technologies 4 Schools (ST4S) framework, which provides a national standard for assessing the security and privacy of edtech products [1]. These frameworks may lead to a "culture of compliance" rather than a proactive adoption of best practice, and their effectiveness is questioned in the Australian context, characterised by its federated education system.

B. Technical Analysis of Children's Apps

Our study combines both static and dynamic analysis. Static analysis examines an app's code to identify potential risks, such as embedded trackers, while dynamic analysis inspects the app's live network traffic to confirm whether data is actually transmitted. This combined approach has been widely used in prior longitudinal studies.

Reyes et al. [8] ran one of the first large-scale dynamic studies on COPPA compliance and found that, of the 5,855 children's apps analysed, a majority were likely to be in violation of the law due to the use of advertising and analytics SDKs. They found that 59% of the apps tested transmitted the AAID, often in conjunction with non-resettable identifiers, and 40% transmitted sensitive data over insecure (non-TLS) connections. Follow-up studies found similar widespread issues: for example, Sun et al. [9] ran a 2021 dataset and found that, among "Family" apps, over 81% used one of the trackers not allowed by Google's policies.

More recently, Alomar et al. [15] offered an important update based on 2023 data and demonstrated the effect of stronger platform policies. They found a "drastic decrease" in violations: specifically AAID transmission dropped from 59% down to 8.8%, and transmission of non-TLS data dropped to less than 1%. However, they concluded that the improper use and misconfiguration of third party SDKs was the main source of privacy risks. This finding is consistent with those of smaller-scale traffic studies of Android and iOS apps [10], [16]. Our work extends this line of technical auditing significantly, through a refined methodology to a new and specific corpus of educational apps.

C. Challenges and Concerns of EdTech Ecosystem

The use of technology in schools is not benign. The large-scale uptake of platforms provided by major technology companies such as Google and Microsoft has been termed the "Googlization" of education and has resulted in dependencies being formed and corporate values of efficiency and productivity being inscribed into pedagogy [18]. These platforms have a business model of "surveillance capitalism" [19] in which the commodity is student and other user data. While many services may be provided to schools for free or for a nominal fee, for the back-end business model of these platforms, it is common to collect massive amounts of student data for commercial purposes [18], [20].

This raises challenges for schools, which typically lack the capacity to undertake effective privacy and security assessments [1], [20]. This gap in resources can result in an "overtrust" in both big tech and the privacy claims of small edtech vendors. In addition, the responsibility for compliance can be spread. While the GDPR places the responsibility with the school as data controller, schools may in turn delegate this trust and responsibility to edtech vendors, creating gaps in accountability [21], [20].

Many countries have established formal assessment or guidance frameworks to examine the privacy and security of edtech used. In Australia, ST4S provides privacy and security asssessment of digital products that target children and schools, and publishes guidance to vendors [22], [23], [24]. The eSafety Commissioner's Safety by Design program prescribes design-level safety principles and has produced toolkits for schools [25], [26]. There is also a Framework for Generative AI in Schools, that defined six principles and 25 guiding statements for the use of safe and ethical AI in schools [27].

In the UK, the Age-Appropriate Design Code requires child-targeting services to consider the best interests of children when designing the default settings [28], and the Department for Education Cyber Security Standards inform security baselines to schools [29]. In the US, COPPA serves as the guideline for child-directed software services. It states that services need verifiable parental consent, and should treat persistent identifiers as personal information [30], [31]. Schools also operate under FERPA and PPRA which are designed by the US Department of Education's Student Privacy Policy Office. There are also voluntary/third-party programs like the Student Privacy Pledge and Common Sense Privacy Program that evaluate practices [32], [33]. In Japan, the Ministry of Education issues an Education Information Security Policy Guideline for schools to develop security policies [34].

These frameworks, although well established, present several issues. For example, most do not require independent validation of runtime telemetry across real devices or realistic usages. In terms of the policy texts, they often are difficult to access and lack plain-language summaries, with no regard to the average education level of parents in their respective populations. The policies also focus towards different perspectives, and not all are compulsory, therefore producing "grey areas" for the apps not specifically targeting children.

D. The HCI Perspective on Children's Privacy

While our work conducts a technical audit of privacy practices, recent HCI literature highlights the user-centric challenges in this domain. Wang et al. [35] identified a disconnect between parents' privacy concerns and their actual management practices, often driven by the unreadability of privacy policies, which is a claim our readability analysis empirically supports. Similarly, Zhang et al. [36] argue that the unclear distinction between child and parent users in app design contributes to privacy vulnerabilities. Our findings on widespread pre-interaction data collection provide the technical evidence underpinning these user-facing challenges.

E. Perspectives of Parents and App Developers

Understanding the privacy landscape requires consideration of the perspectives of the human actors involved. Research into parents' views found a consistent set of top concerns: screen time, inappropriate content, and contact with strangers [21]. Commercial data collection was a concern but less prominent. Parents delegate trust to schools/educational authorities to vet the technologies their children use, assuming that if a app is approved by the school, it is "safe". This delegation of trust and responsibility, combined with a lack of technical knowledge, meant that parents did not typically engage with trust settings or privacy policies [21].

Similarly, several studies have been conducted on children's online-privacy. Qualitative studies show that children often frame privacy mainly as interpersonal control while underestimating institutional and commercial data collection [37], [38]. Studies with young children found that they understood data collection in terms of its immediate benefits (e.g. saving their progress in a game), and viewed privacy in an interpersonal way (i.e. keeping a secret from people) rather than an institutional one (i.e. corporate data collection) [9]. Their understanding was driven by surface-level visual cues in the interface of an app, and data collection that occurred invisibly in the background was, conceptually speaking, non-existent to them. This shows that children's own conceptualisations of privacy are still developing.

Prior work on children's developing privacy literacy, and ecosystem level risks in child-directed apps, motivate our focus on minors and our measurement of policy–behaviour alignment, where such works propose concrete, age-appropriate learning objectives to help youth reason about data flows across contexts. [39] Recent dyadic interviews with youth–parent pairs further reveal where parental guidance helps (and where gaps persist), underscoring that privacy and security knowledge co-evolves within families [40]. On the ecosystem side, app-focused evidence shows persistent risks: a scoping review finds many child-related apps exhibit weak privacy/security practices and manipulative commercial features [41], while a 2023 traffic analysis documents extensive data sharing among children's iOS apps [16]. Together, these studies justify a dedicated subsection on children and motivate our own measurement of policy–behaviour alignment in apps targeting (or routinely used by) minors [42].

On the other side of the app ecosystem, the same is true for developers. Studies engaging with developers directly have shown that, while most express a desire to protect children's privacy, they are limited by powerful systemic factors [43]. The main constraint is the prevailing business model of the app economy, which is heavily advertising-based. Developers report that monetisation options other than ads are not financially viable due to competition in the app store and users' expectations of free content [43]. As a result, they are forced to incorporate third-party ad SDKs, even if they are aware of the privacy costs. In addition, developers struggle with a lack of clear, actionable design guidelines, and find the landscape of third-party libraries opaque and difficult to navigate, defaulting to popular (but data-intensive) libraries from the big tech companies.

The previous works mentioned above primarily rely on static analysis or single methods. In contrast, our study attempted to combine four types of examination: Static inspection on APKs; Dynamic runtime analysis; policy-behaviour alignment checks and readability analysis of the privacy policies. Unlike previous studies, where the corpus of apps are obtained from one single store, our corpus is collected from various sources, including Australian school websites, educational department websites, and the Schools Catalogue Information Service. Therefore, our results correlate strongly to the Australian context. We also investigate further from, for example, the question of whether trackers are present or not, to the examination of when do data flows occur; what identifiers are sent, and who receives them.

III. Data Collection

To ensure our analysis is both comprehensive and replicable, we developed a multi-stage data collection procedure, involving static and dynamic analysis of the apps, as well as the analysis of the privacy policies.

A. App Corpus Curation

We collected a list of 200 unique child or student-focused Android applications relevant to Australian children [44]. To achieve a sample that reflects the apps formally used in education, such as the ones endorsed by relevant authorities, and those available on the general market, we used an elicitation distribution as follows.We collected these applications from a range of publicly accessible and educationally endorsed sources, including State Department of Education websites[2] , the Schools Catalogue Information Service (SCIS)[3] , the Google Play Store[4] , and Australian schools' official websites listing recommended educational apps. More detailed distribution can be seen in Table I. The categories of the applications were modified based on Victoria's official catalogue *Arc*[7], with minor changes to facilitate the diversity of applications within our corpus.

[2]For example: NSW Department of Education — https://education.nsw.gov.au ; Victorian Department of Education — https://www.education.vic.gov.au

[3]https://www.scisdata.com

[4]https://play.google.com

TABLE I
APP COUNTS BY PRIMARY CATEGORY AND SOURCE

Category	Count	Category	Count
Literacy/ELA	36	Edutainment/Game	15
Math	26	Coding/Robotics	9
Classroom Tools	20	Reference/Dictionary	8
Music/Art	16	Assessment/Quiz	5
Science/STEM	19	Communication	3
Library/eBooks	9	Social Studies	3
Languages	16	LMS/Portfolio	1
Wellbeing/SEL	11	Utilities	4
Source of Apps			
SCIS	40 (20.0%)	School Sites	56 (28.0%)
Depart. Sites	35 (17.5%)	Play Store	69 (34.5%)

B. Data Collection Pipeline

As stated, for the educational applications we examined, the corpus covers popular learning tools recommended by various Australian secondary schools and educational departments, listings in the Australian SCIS, and high-ranking "Kids" and "Educational" apps from the Google Play Store. We gathered the set of analysis with the following pipeline illustrated in Figure 1.

1) Static Analysis: We first collected static data using the Mobile Security Framework (MobSF) [45] to generate reports for each app. MobSF is an open-source toolkit for mobile app security testing. For Android APKs, it performs static, source-less analysis by decompiling the package and extracting manifest and code artefacts. From this, it enumerates requested permissions, embedded third-party packages/SDKs and common issues such as hardcoded API keys/tokens. In reports, we could identify potential risks by listing embedded third-party libraries, the requested permissions in the app's `AndroidManifest.xml`, as well as security vulnerabilities such as the hardcoded API keys. We also collected the corresponding privacy policies of the apps from their websites.

2) Dynamic Analysis: We then performed dynamic analysis where each app was installed via the Google Play store on a rooted Pixel 8a Pro emulator running on a Windows 11 computer. We employed Burp Suite to act as the Man-in-the-Middle proxy to intercept and decrypt the outbound HTTPS traffic from the emulator. Some applications use SSL pinning, which potentially could prevent Burp Suite from capturing the traffic, as the application would not trust Burp's certificate. We employed PCAPdroid[5], an on-device proxy tool that can bypass this issue. Inspired by differential analysis methodologies [46], we generated 2-3 distinct traffic logs per app, to capture behaviour in different contexts:

- **Stage 1 (S1):** We recorded approximately 5 minutes of network traffic immediately from the app's first launch after installing with no user input. We chose 5 minutes window, as we noticed in early experimentations, that first-run SDK and tracker initialisations (e.g., config fetches, token exchanges, etc.) typically start within the first 3 minutes of the app's launching.
- **Stage 2 (S2):** We simulated user interaction by opening the app and manually navigating through it for 5-10 minutes, intentionally accessing features that appear to be related to privacy, such as account settings, privacy settings, etc., We also explored normal functionalities of the apps, such as the games or lessons, thus imitating the behaviour of a normal adult user, such as a parent or teacher.
- **Stage 2b (S2b):** If the app allows two types of logins e.g., child and parent/teacher accounts, we revisit it to imitate the behaviour of a child user.

After capture, each app's runtime traffic was exported as either a Burp XML HTTP history (when TLS interception via Burp Suite succeeds) or a PCAP file (when we switch to PCAPdroid to bypass SSL pinning). We kept both formats because they originated from different toolchains; for analysis, we parsed them into a single, normalised schema so downstream checks were identical regardless of source. For each app, we retained up to four artefacts: i). The S1 and S2 logs; ii). S2b log if applicable; (iii) the Stage 2b (S2b) child-behaviour log when a child account path exists, and (iv) the MobSF static report.

To discover the discrepancies between what the static reports claim and what the dynamic logs reveal, we integrated both findings in three steps. First, we mapped each contacted domain (e.g., app-measurement.com, sentry.io) to its SDK/service family. Second, we cross-checked whether those SDKs were present in MobSF outputs (packages, classes, manifests). Third, we annotated identifier/timing events, e.g., co-occurrence of AAID and FID/Installation ID, and whether transmissions occurred during idle (S1) vs interactive (S2) or in child (S2b) sessions. We used LLMs (Gemini 2.5 Pro, ChatGPT-5) to assist in analysing the dynamic logs and static reports, and producing reviewer-facing summaries.

3) Privacy Policy Analysis: We then proceeded to source the privacy policy texts of the tested applications, by manually searching and going through the official websites of the tested applications or the developers, then saving the links to the policies in an excel sheet. Some of the applications / developers did not have an official website and therefore we could not capture the policies for those. We then used these links to access the policies, and saved them in .txt files. We used multiple python libraries to examine the readability of the policy texts, for example, ranking with the Flesch-Kincaid test, SMOG index and Gunning Fog. We also compared the policies with the dynamic logs to check discrepancies between the traffic and the policy.

IV. RESULTS

In this section, we explain the empirical findings across the four types of analysis: static and dynamic analysis, policy-behaviour alignment and readability.

[5]https://emanuele-f.github.io/PCAPdroid/

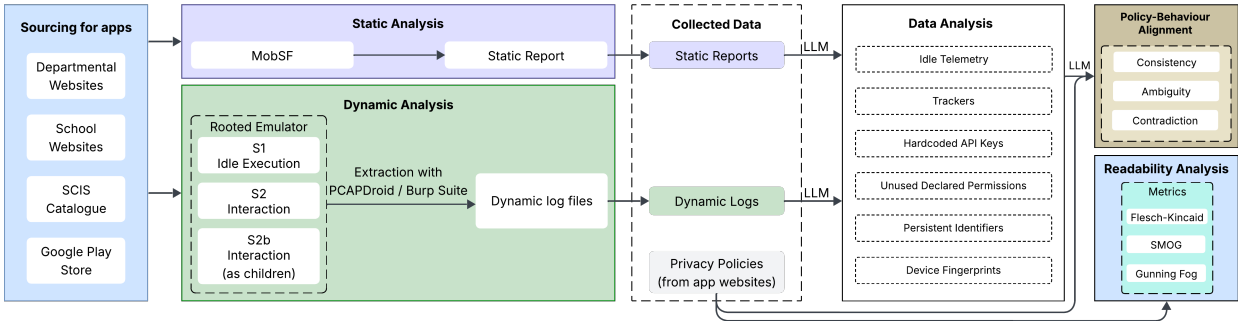

Fig. 1. Data Collection Pipeline

A. Static Analysis

1) Third Party SDKs: Our static analysis reveal that there is widespread embedding of third-party components, with 67.9% of applications containing at least one identifiable tracker or analytics SDK, such as `Firebase`, `AdMob`, `Unity Analytics`, `Facebook SDK`, etc. Across our corpus, the density of third-party analytics/ads integrations is modest: we observe a mean of 2.65 static tracker SDKs per app and 2.68 distinct runtime tracker domains. The top 10 most declared trackers are shown in Table II.

TABLE II
TOP 10 STATIC SDK/TRACKER FAMILIES IN APKS

SDK/Tracker	Apps (count)	% of Apps
`Firebase`	125	62.6
`Crashlytics`	67	33.5
`Facebook SDK`	49	24.5
`Unity Analytics`	32	16.1
`Google Ads`	30	14.8
`AppsFlyer`	23	11.6
`RevenueCat`	15	7.7
`Adjust`	12	5.8
`Sentry`	9	4.5

TABLE III
TOP HARDCODED KEY/TOKEN AND SUBTYPES

Item	Apps (count)	% of Apps
Any hardcoded key/token	158	79
`Firebase`	64	32
`Google` x-goog-api-key	22	11.0
`Crashlytics` token/key	17	8.5
`RevenueCat` key / token	9	4.5
`Google` API key (generic)	7	3.5

Notes: "Any hardcoded key/token" counts an app once if *any* secret subtype is present. Subtypes are not mutually exclusive; therefore, subtype counts need not sum to the "Any" total.

2) Hard-Coded API Keys: Hard-coded API keys or tokens refer to credentials that are directly embedded within an application's code, such as in manifests, resource files, or string constants compiled into classes. Because APKs are easily decompiled, these secrets can be extracted, exposing the application to several security risks, including, i) unauthorised use of paid APIs or quotas leading to cost or fraud, ii) privilege escalation against a vendor's backend if the key provides access to sensitive endpoints, iii) account takeover of telemetry systems (e.g., falsifying crash or analytics data), and iv) cross-application compromise when the same key is reused across products or environments. As shown in Table III, our results indicate that 79% of the analysed apps contained such hard-coded API keys or tokens—most commonly `Google` API keys found within `Firebase` or `Analytics` headers. The most prevalent secret types are `Firebase` API keys (64/200; 32.3%) and `Google` x-goog-api-key (22/200; 11.0%).

In Table IV, we demonstrate how prevalent are hard-coded secrets in each category of app. The highest rate appears in Library/eBooks (9/9; 100%), Math (24/26; 92.3%), Languages (15/16; 93.8%), Wellbeing/SEL (10/11; 90.0%) and Edutainment/Game.

In terms of the number of secrets per app, the most typical case , i.e. the most frequent pattern is a single exposed key (typically `Firebase`) used for analytics/configuration, observed in 138/200 apps (69.0%), however, about 1 in 10 apps (20/200; 10.0%) embed multiple secret types.

The drastic difference between categories could be attributed to the behavioural differences of the apps. For example, Library and Reference apps exhibited the highest percentage of exposed secrets (100%), which is likely due to their reliance on external cloud services to retrieve dynamic content, leading developers to embed cloud storage credentials directly into the client-side code to assist with data fetching. In contrast, music and art apps typically function as offline tools to compose music or paint pictures which require minimal connectivity.

3) Sensitive Information: 39.4% of apps requested sensitive permissions that were never utilized during dynamic testing. The most common 'oversearched' permissions were CAMERA (16.8%), RECORD_AUDIO (14.2%), and WRITE_EXTERNAL_STORAGE (13.5%).

Research Takeaway: Static analysis reveals a "risk-by-design"footprint. Even before an app is run, the widespread embedding of third-party trackers (67.9%) and hard-coded secrets (79%) creates a large, latent attack surface. This, combined with over-privileged permission requests (39.4% unused sensitive permissions), suggests that developer convenience and reliance on third-party libraries often override

TABLE IV
OVERALL DISTRIBUTION OF HARD-CODED SECRETS BY CATEGORY (APPS WITH ≥1 SECRET; CATEGORIES WITH ≥5 APPS)

Category	Apps (n)	With Secret (n)	Prevalence (%)
Library/eBooks	9	9	100.0
Languages	16	15	93.8
Math	26	24	92.3
Wellbeing/SEL	11	10	90.0
Edutainment/Game	15	13	86.7
Reference/Dictionary	8	7	87.5
Science/STEM	19	16	84.2
Coding/Robotics	8	6	75.0
Classroom Tools	20	15	75.0
Literacy/ELA	36	24	66.7
Music/Art	16	9	56.3

the principle of data mimimisation.

B. Dynamic Network Analysis

Dynamic testing confirmed that the majority of apps transmit data almost immediately after installation. 89.3% of applications generated outbound traffic before any user interaction, for example, contacting `Firebase` Installations, `Google Analytics` or `AdMob` endpoints within seconds of launch. 83.6% transmitted at least one persistent identifier (e.g. `Firebase` Installation ID, `Advertising` ID or `Crashlytics` Installation ID). 86.4% apps disclosed some form of device fingerprints.

1) Idle Traffic - Stage S1: As seen in Table V, a striking 89.3% of applications initiated outbound connections before any user behaviour, sometimes within the first one to two seconds after installation. This "idle telemetry" usually targets analytics destinations such as `firebaseinstallations.googleapis.com`, `app-measurement. com` or `config.uca.cloud.unity3d.com`. Several apps, including *Math Lingo* and *Makers Empire 3D* began exchanging configuration packets while still displaying the splash screen, well before any privacy prompt of terms-of-service acceptance appeared.

From a consumer perspective, early network connections may appear to be standard 'app startup' behaviour. However, our findings show these connections are not for functional assets but for telemetry and profiling. Under Australian Privacy Principles and COPPA, data collection should be governed by informed consent; when 89.3% of apps transmit data before a user can even open the privacy policy, the 'notice-and-consent' framework fails entirely. This effectively creates a shadow handshake that identifies and profiles a child before they or their parents have any agency in the process.

2) Interactive Traffic - Stage 2: During S2, most apps continued the S1 identity and expanded with first-party feature toggles and telemetry frameworks. For example, `Sentry` and `Amplitude` logged feature-use and monetisation events (e.g., paywall impressions/flows) with SDK metadata, and sent exception/performance envelopes during user flows. 96% apps contacted at least one external domain during the testing, and most of these domains correlate closely to Google (`Firebase/Crashlytics`, etc.), `Unity`, `Meta`, `Amplitude`, etc. In terms of identifiers and fingerprints, S2 typically develops on top of the S1 baseline (e.g., app version, feature flag).

Using the same app-category taxonomy as elsewhere in the paper, we computed S2 ("interactive use") metrics by category (Table VI). Contact with analytics endpoints is most prevalent in Assessment/Quiz (100%), Math (92.9%), and Library/eBooks (83.3%). Advertising/marketing endpoints are most common in Music/Art (30.0%), Literacy/ELA (24.0%), and Edutainment/Game (20.0%). Persistent identifiers (AAID, FID, or comparable device IDs) are widespread, with Assessment/Quiz (100%), Coding/Robotics (100%), and Science/STEM (94.1%) showing the highest rates. As a linkability indicator, we report the share of apps where AAID and FID both appear somewhere in the dynamic record (e.g., Music/Art 90.0%, Science/STEM 94.1%, Wellbeing/SEL 87.5%).

As seen in Table VI and VII, we stratified S2 behaviours by subject category and child-facing branding (titles containing key words such as Kids/Preschool/ABC/Phonics/Jr). Telemetry remains prevalent in both groups (kids 78.9%, general 77.7%). Ad-tech signals are somewhat less frequent in child-facing titles (kids 42.1%, general 48.5%). Explicit S2 mentions of persistent identifiers appear at low single-digit rates (kids 5.3%, general 7.8%), and location references are rare.

3) Child-Behaviour - S2b: We treat S2b as a separate user on a new installation, not a continuation from S2, therefore using a separate column in Table V. We gathered 20 apps that featured particular options to register as an underage user. In S1+S2, identity is already established at idle for most apps (global: identifiers in 83.6%; idle transmissions 89.3%). S2b goes further for half the titles, where we see a new `Firebase` Installation (new FID) or a new auth token for the same FID at the start of the child path. This indicates that instead of only continuing an existing analytics identity, S2b creates or refreshes the identifier set for a second user, multiplying identity records per device and widening linkage possibilities across runs/users on the same handset. We see that less apps in S2b expose hardcoded API keys, but the percentage of apps transmitting persistent identifiers remained relatively similar. In these apps with child-account options, we observe that more would embed tracker SDKs, as well as leaking device fingerprint fields.

These third-party SDKs introduce concrete privacy–security risks for children's apps. Analytics SDKs (e.g., `Firebase Analytics`, `Amplitude`) rely on device/installation identifiers (FID, device ID), which are persistent identifiers and therefore "personal information" under COPPA; collection and use outside "internal operations" requires consent and strong governance [30], [47], [48]. Crash reporting SDKs (e.g., Crashlytics, Sentry) can ingest sensitive data via custom logs, keys, and breadcrumbs unless teams actively scrub/disable PII; both vendors document PII-scrubbing controls, and prior large-scale studies show sensitive information commonly appears

TABLE V
SUMMARY OF DYNAMIC ANALYSIS RESULTS BY STAGE

Metric	S1+S2 (N=200)	S2b
Total apps analysed	200	20
Apps embedding ≥ 1 tracker SDK	67.9%	**85%**
Apps exposing hardcoded API keys or tokens	73.6%	**25%**
Apps declaring unused permissions	48.6%	**75%**
Apps generating idle (pre-interaction) transmissions	89.3%	n/a
Apps transmitting persistent identifiers (FID, AAID, etc.)	83.6%	**85%**
Apps leaking device fingerprint fields (any)	86%	**95%**

Note: The 73.6% rate for hardcoded secrets in S1+S2 differs from the 79% previously reported (Table III) as this column isolates findings observed specifically within the combined S1 and S2 interaction logs. The unused permissions here include all permissions, hence higher than the previously reported 39.4%, which only counts for sensitive ones.

TABLE VI
INTERACTIVE TRAFFIC AND RISK MARKERS BY CATEGORY OF APPS (N$\geq$5)

Category	Analytics (%)	Ads/Marketing (%)	Identifiers (%)	AAID+FID (proxy) (%)	Unused sens. perms (%)
Assessment/Quiz	100.0	0.0	100.0	0.0	80.0
Classroom Tools	82.4	11.8	88.2	17.6	100.0
Coding/Robotics	77.8	0.0	100.0	11.1	88.9
Edutainment/Game	73.3	20.0	80.0	20.0	93.3
Languages	72.7	18.2	81.8	18.2	90.9
Library/eBooks	83.3	0.0	83.3	33.3	100.0
Literacy/ELA	60.0	24.0	80.0	24.0	88.0
Math	92.9	14.3	85.7	14.3	85.7
Music/Art	60.0	30.0	70.0	10.0	90.0
Reference/Dictionary	50.0	0.0	87.5	12.5	100.0
Science/STEM	64.7	5.9	94.1	35.3	94.1
Wellbeing/SEL	75.0	12.5	87.5	25.0	87.5

Group	Telemetry (%)	Ads/AdTech (%)	Persistent IDs (%)	Location (%)	Median runtime domains
Child-facing	78.9	42.1	5.3	5.3	1.0
General	77.7	48.5	7.8	0.0	3.0

TABLE VII
INTERACTIVE TRAFFIC (S2) BY BRANDING OF APPS.

in logs [49], [50], [51]. Ads/attribution SDKs depend on advertising or device identifiers for cross-app measurement; Google Play's Families Policy permits ads to children only via self-certified ad SDKs and imposes extra restrictions for mixed audiences [52].

4) Persistent Identifiers: We observed app- and device-scoped identifiers that persist across sessions, such as the `Android` Advertising ID (AAID/GAID)—a user-resettable, device-wide ad identifier; the `Firebase` Installation ID (FID), an app-instance identifier used across `Firebase` services; and the `Crashlytics` installation UUID, which is an app-instance identifier used for crash reports. Roughly 83.6% of apps transmitted at least one persistent identifier, most commonly the `Firebase` Installation ID (FID), Advertising ID (AAID), or the `Crashlytics` Installation ID. These identifiers are nominally pseudonymous but, when combined across SDKs (e.g., an attribution SDK receiving its own `AppsFlyer` ID alongside the device's Android Advertising ID), they allow the SDK operator to recognise the same device across sessions and across other apps that embed the SDK. Under COPPA, such use of persistent identifiers is treated as collection of personal information over time and across online services that can enable cross-session and cross-app linkage.

Traffic logs revealed multi-identifier transmissions within the same POST payloads, a clear indication of SDK-level correlation potential. For instance, *Quizlet* and *Seesaw* transmitted both a `Firebase` ID and an `AppsFlyer` Install ID, providing two orthogonal user anchors that third-party processors can reconcile. This means that the FID enables `Firebase` to link analytics and crash reports within a single app installation, whereas the AppsFlyer ID, often transmitted alongside the device-level GAID/AAID, allows `AppsFlyer` to recognise and track the same device across multiple apps that include its SDK. When several persistent identifiers are sent together in one request, each provider gains a durable, joinable key for that installation and, when GAID/AAID is present, the capability for cross-app tracking. Only a small minority (16%) confined themselves to ephemeral session identifiers.

Additionally, 86% apps leaked detailed device fingerprints, such as OS version, model string, locale, carrier name and build version. This fingerprint data is

often sent in JSON payloads like: `"model":"Pixel 8 Pro", "locale":"en_AU", "density":2.625, "osVersion": "Android 14"`. Such parameters, while seemingly benign, when transmitted alongside other persistent identifiers such as the advertising IDs, can assist in maintaining stable user linkages as observed by Reyes et al. (2018) [8].

5) Runtime Domains: Runtime domains are the external internet hosts an app talks to while it runs. There are two types: first-party vendor APIs and third-party SDK backends (analytics, crash, attribution). Runtime domains matter for security and privacy because each extra external endpoint expands the attack and trust surface and can expose more user data.

Our analysis reveal that apps contact remote hosts almost universally. 96% of apps reached at least one external domain during the test sessions, while only 4% showed no observable endpoints and no identifiers/fingerprints. Among the apps that did connect, the destinations remain highly centralised. The most frequently observed domains across the corpus are Firebase endpoints (e.g., `firebaseinstallations.googleapis.com`, `firebaselogging-pa.googleapis.com`, `Crashlytics settings`), Facebook Graph (`graph.facebook.com`), `Unity config` (`config.uca.cloud.unity3d.com`), `Google Play` (`play.googleapis.com`), `app-measurement.com`, and `RevenueCat` (`api.revenuecat.com`). The fact that 96% of apps reach a small set of third-party analytics/crash/attribution providers (e.g., `Google/Firebase & Crashlytics`, `Meta/Facebook SDKs`, `Unity Analytics`, `Amplitude`, `Sentry`) indicates a concentrated telemetry infrastructure: a few SDK operators receive data from many unrelated apps. Prior work shows this ecosystem is long-tailed but dominated by a handful of trackers, enabling broad cross-service visibility for those providers [53]. This centralisation has broad implications. Firstly, when these SDK endpoints receive persistent identifiers, they can recognise the same user/device over time and across services. Secondly, third-party libraries expand the attack and trust surface. Empirical security work showed Android SDKs can introduce vulnerabilities or misuse inherited permissions, amplifying privacy risk for host apps [54].

Research Takeaway: Taken together, the runtime findings depict an ecosystem where data disclosure precedes user agency. Almost 9 in 10 apps initiate telemetry on boot, 5 in 6 maintain long-lived identifiers, and almost 7 in 8 export a granular hardware profile. The median educational app therefore operates closer to a consumer analytics client than a classroom utility. These patterns, observed even among apps promoted on state education portals, reveal a privacy model that depends on user passivity and institutional trust, not on genuine data minimisation.

TABLE VIII
ALIGNMENT DISTRIBUTION BY TARGET AUDIENCE

Target Audience	**Conflict (%)**	**Partial (%)**	**Consistent (%)**	**Other (%)**
Child-targeted	38.1	38.1	23.8	0.0
General-audience	34.0	33.0	25.8	7.2

TABLE IX
POLICY COVERAGE BY CATEGORY.

Category	Coverage (%)
First-Party Collection/Use	90.4
Third-Party Sharing/Collection	87.8
User Choice/Control	87.0
Data Security	79.1
Data Retention	74.8
Policy Change	72.2
Privacy Contact Information	71.3
Practice Not Covered	69.6
Introductory Generic	67.8
User Access, Edit and Deletion	67.0
International & Specific Audience	55.7
Do Not Track	39.1

C. Policy-Behaviour Alignment

1) Policy coverage by OPP-style categories: We code policy paragraphs using an OPP-style taxonomy derived from the OPP-115 corpus, which organises privacy policy text into 12 top-level, end-user-oriented categories (e.g., First-Party Collection/Use, Third-Party Sharing/Collection, User Choice/Control, Data Security, Data Retention, User Access/Edit/Deletion, Do Not Track, International & Specific Audience, etc.). In line with recent OPP-based classification work that targets these 12 classes for paragraph-level labelling [55], we operationalise coverage as a binary flag per app indicating whether the policy contains at least one statement in a given category. Policies with extraction failures (all zeros across categories) are excluded from denominators.

Table IX reports the share of apps mentioning each category. Coverage is highest for foundational topics—First-Party Collection/Use (90.4%), Third-Party Sharing/Collection (87.8%), and User Choice/Control (87.0%)—suggesting most policies acknowledge core data flows and some notion of user agency. Mid-tier coverage appears for Data Security (79.1%) and Data Retention (74.8%). Lower attention is paid to International & Specific Audience (55.7%) and especially Do Not Track (39.1%), indicating that cross-border issues and DNT signalling are less consistently addressed in policy text.

We next show that the comparison between developers' declared privacy practicies and their apps' observed network behaviour revealed a striking and perhaps systematic pattern of misalignment. Although privacy policies are intended to provide transparency and consent foundations under the Australian Privacy Principles and comparable child-data regimes, the majority of the sampled policies failed to describe what the applications actually did. This misalignment is not an isolated occurrence; rather, it appeared to be structural, and measurable.

2) Overall Distribution: Out of the policies analysed, only a fractional minority exhibited direct textual consistency with empirical evidence. We classify the behaviour of applications into five types, as described in Table X.

These proportions reveal that nearly half of the policies only appear compliant on the surface, while roughly another one-quarter engage in partial or misleading disclosure. Less than 2% of developers could be verified as entirely first-party and telemetry-free.

3) Patterns of divergence: The data shows three recurring types of misalignment, each representing a different failure:

(a) Partial-disclosure of analytics frameworks: Even when policies admitted "usage analytics", 25.4% of those cases failed to name the specific SDKs involved, as seen in Table X. For instance, *Maker's Empire* declared the use of "anonymous analytics for service improvement", but dynamic inspection uncovered `Firebase Analytics`, `Amplitude`, and `RevenueCat` concurrently sending session-linked identifiers to multiple third-party domains. Such omissions undermine meaningful consent because parents or teachers cannot know which entities process the data.

(b) Contradiction: A significant group of apps, around one-fifth (20.8%)—explicitly claimed no personal data collection, but as seen in Table X, still transmitted persistent installation or advertising identifiers within seconds of startup. *Matific*, for example, stated "no ads, no tracking", but runtime logs showed `Unity` Analytics and `Google` Favicon requests initiated before user interaction. Similarly, *Merriam-Webster Kids* sent both `Firebase` Install IDs and `AdMob` telemetry less than three seconds after launch, despite a child-directed declaration of "limited collection". Three detailed examples can be seen in table XI.

(c) Ambiguous: Many policies relied on general, vague compliance text such as *"We may collect technical data including device type for debugging purposes"* without specifying collection frequency, destination, or persistence. This language creates the illusion of being compliant while offering no practical transparency. The presence of *may*, *might* and *such as* clauses correlate strongly with behaviour-policy misalignment: 82% of policies with these phrases are associated with apps that transmitted persistent IDs.

4) Examples Cases: The divergence becomes clearer through examples like below that contrast the apps' written disclosures with observed runtime behaviour such as identifiers, timing, recipients, and security, to show where alignment holds or breaks. Prior work shows that dynamic traffic often reveals undeclared collection/sharing—especially of persistent identifiers that qualify as personal information under `COPPA`; therefore, these contrasts make divergence concrete [8].

- *Animal World* (strong alignment): Consistent – observed telemetry (`Unity Analytics` and `Firebase` logging) matches declared third-party services (`Unity`, `Google`); no undeclared SDKs or ad networks detected.
- *Phonics Hero* (Partial alignment): Partial – declared vendors match most observed functions, but CreateJS and CloudFront SDKs used for gameplay assets are not mentioned in the policy.
- *CamScanner* (Conflict): Inconsistent – network logs show heavy third-party ad and tracking SDK activity (`Appsflyer`, `ByteDance`, `Facebook`, `Unity Ads`, `AppLovin`) exceeding what policy discloses as "limited sharing"; policy minimises scope of ad data.

Research Takeaways: A closer inspection of app titles reveals that child-branded software does not equate to safer privacy behaviour. Using keyword cues such as kids, preschool, ABC, and phonics, 21 applications in the corpus were identified as child-targeted, which directly reflects how these apps were identified, while 158 were classified as general-audience educational. Despite their explicit child focus and placement in *Kids* or *Early Learning* categories, these titles displayed comparable—if not slightly worse—policy–behaviour alignment outcomes than their general counterparts.

5) Deceiving Names: As observed in Table VIII, two observations stand out.

First, 76% of the child-targeted apps (Conflict + Partial) exhibited at least one form of misalignment—undeclared SDKs, contradictory "no-data" claims, or ambiguous disclosures, which is considerably higher than the 67% rate among general educational titles. Second, only one in five child-targeted apps could be verified as fully consistent with their stated privacy commitments, compared with about one-quarter of general apps.

The qualitative descriptions reinforce this pattern. Apps with overtly child-friendly branding—*Merriam-Webster Kids, Matific, Maker's Empire*, *ABC Phonics*—were routinely coded as Partial or Conflict due to undisclosed Firebase, Unity, or AdMob telemetry. By contrast, general-audience tools such as *Khan Academy* or Quizlet were more likely to name their analytics providers explicitly, suggesting more mature privacy governance frameworks.

These findings illustrate what might be termed the illusion of safety: labels like *Kids*, *Preschool*, or *Educational for Children* cultivate parental trust and are often associated with school recommendations, yet they do not correspond to stronger technical or policy compliance. Instead, they may mask legacy SDK integrations or inherited ad modules, reproducing the same data-sharing behaviours seen in commercial entertainment apps.

From a policy perspective, this insight challenges a persistent regulatory assumption—that child-directed categorisation ensures enhanced protection. The evidence here suggests the opposite: child-facing branding is not a reliable proxy for privacy assurance. Consequently, educators and procurement bodies relying on "Kids" category listings as an implicit compliance filter risk endorsing applications that transmit identifiers and analytics data at the same rates as general-audience tools.

D. Policy Readability

Beyond examining how well the behaviours of apps align with their privacy policies, we also evaluated the interpretabil-

TABLE X
POLICY–BEHAVIOUR ALIGNMENT OVERVIEW

Category	Share (%)	Description
Consistent declared alignment	49.1	Policy and telemetry broadly consistent; declared SDKs match observed domains.
Partial disclosure	25.4	Policy admits analytics but omits specific SDK names or processors.
Conflict / contradiction	20.8	Claims of "no personal data collection" contradicted by telemetry containing identifiers.
Ambiguous / mixed	2.9	Language partially consistent but internally contradictory or vague.
Strong compliance / no trackers	1.7	Verified first-party-only behaviour; no third-party endpoints detected.

TABLE XI
EXAMPLES OF "CONTRADICTORY CLAIMS" (POLICY/STORE CLAIM VS. OBSERVED RUNTIME IDENTIFIERS)

App (dataset)	Public claim (policy/store)	Observed at startup (S1)	Why contradictory
Merriam-Webster Kids	Children's/Kids product line; corporate policy says it does not knowingly collect children's personal data and positions kid-oriented use	FID and AdMob initialization within ∼3s of launch; telemetry persists in S2	Immediate creation/use of persistent identifiers despite child-facing positioning
Women Who Changed the World (Learny Land)	App Store listing marked "Data Not Collected"; developer directs to a privacy policy claiming no personal data from children	Firebase installation/analytics initialization at first run; analytics events continue during S2	"Data Not Collected" label conflicts with persistent ID/analytics initialization
Bugs and Numbers (Bugs series)	App Store privacy section marked "Data Not Collected" for the title in the Bugs series	Startup traces show Firebase initialization and continuing entitlement checks during S2	Public "no data collected" claims are inconsistent with persistent identifier setup

ity of the policies using four standard metrics: i) Flesch-Kincaid Reading Ease (FRE), ii) Flesch-Kincaid Grade Level (FKGL), iii) SMOG, and iv) Gunning Fog Index. 34 apps were excluded from the analysis due to inaccessibility. In this analysable subset, FRE had a median of 32.43; FKGL a median of 14.81, SMOG had mean 16 and Gunning Fog mean 17.81. Only four policies (3%) achieved an FRE above 50, and only one exceeded 60. Overall, the results suggest that the typical privacy policy requires tertiary-level literacy, far above the average Australian adult's reading ability. For instance, in FRE, a score above 50 is considered "fairly easy" to read. In other words, an average parent would need a university-level education to comfortably comprehend these texts.

This readability gap demonstrates that, with the previously mentioned transparency deficit in section IV-C, even when policies do disclose analytics or data collection, the information is often presented in a language manner exceedingly complex to serve its intended reader. This aligns with prior observations that child-relevant services frequently write vague or inconsistent disclosures [46]. In parallel, platform and legal institutions require accurate and accessible disclosures and, for many processing purposes, verifiable parental consent [15][9]. Google Play's policies expressly require developers to provide links to privacy policies and warn that inaccurate disclosures are "deceptive" [15]. At law, simply notifying parents via a policy is not sufficient, as COPPA an GDPR require verifiable parental consent before collecting children's data, especially when identifiers are shared with third parties [9].

In Australian schools, there are acknowledged limitations to the consent-centric privacy mechanism and difficulty for schools in assessing vendors' data practicies [1]. Consequently, clearer, more readable policies are likely to improve administrators' ability to verify claims against technical behaviour and to implement appropriate consent flows and controls, for example, SDK configurations and third-party tracking.

It is not impossible to construct accessible, "plain-English" policy texts, as demonstrated by the few examples in our analysis; however, unfortunately they are not the common practice currently. Developers can improve scores, for example, by shortening sentences and reducing "legalese".

V. DISCUSSIONS

This study analysed the Australian educational app echosystem from four perspectives: static code, dynamic traffic, policy-to-practice alignment, and policy readability. The findings paint a concerning picture of an ecosystem that operates on a foundation of implicit trust while engaging in risky and opaque data practices. Our results can be synthesised into three primary themes: a "risk-by-design" development culture, the "illusion of safety" created by child-centric branding, and the structural impossibility of informed consent.

A. Risk-by-Design and Non-Consensual Collection

Our results point to a risk-compliant culture in app development. The static analysis revealed that a majority of apps embed third-party trackers (67.9%) and, alarmingly, hard-coded API keys (79%), creating a latent attack surface before an app is even launched. This risk is immediately observed upon launch, as shown by our dynamic analysis. The finding that 89.3% of apps transmit data before any user interaction or consent is a critical finding. This practice, combined with the routine transmission of persistent identifiers like the Firebase Installation ID (FID) by 83.6% of apps, demonstrates that data collection is not an opt-in choice but a non-consensual prerequisite for participation.

B. Illusion of Safety

Perhaps the most striking finding is the "illusion of safety" created by child-centric branding. Our policy-behaviour alignment analysis revealed that apps explicitly branded for "Kids," "Preschool," or "ABC" were not safer than their general-audience counterparts; in fact, they were less likely to be consistent with their own policies (23.8% consistent vs. 25.8% for general apps). A significant portion of apps (20.8%) exhibited direct contradictions, such as claiming "no personal data collection" while actively transmitting persistent identifiers. This suggests that child-centric branding is often a marketing tactic that cultivates a false sense of trust among parents and educators, rather than a genuine indicator of enhanced privacy or technical compliance.

C. Failure of Informed Consent

Even if an app's practices were perfectly aligned with its policy, our analysis shows that informed consent is structurally impossible. The median Flesch-Kincaid Grade Level of 14.81 means these legal documents require a university-level education to comprehend. With only 3% of policies being "fairly easy" to read or better, the privacy policy serves as a tool for legal defence for the company, not as a transparent disclosure for the user. When this unreadability is combined with "idle telemetry" (data sent before the policy can be read) and deceptive disclosures, the entire "notice-and-consent" framework is shown to be failing.

D. Implications

Taken together, our findings demonstrate a systemic failure. Parents and educators, who are legally and morally obligated to protect children, are being let down. They are forced to rely on deceptive branding and unreadable legal documents, all while apps silently transmit persistent identifiers to a centralised group of third-party SDKs (e.g., Google/Firebase, Meta, Unity). This aligns with findings from Pangrazio & Bunn (2024), who identified a "technology overtrust" in Australian schools, with our technical findings provide the empirical evidence for why this overtrust is considerably dangerous. It confirms that the current model of self-regulation, and brand-based trust is placing the burden of privacy on the very users who are least equipped to manage it.

To address these systemic failures, our findings suggest that policy and practice must shift from a reliance on self-regulation toward proactive enforcement and technical verification. First, the categorisation of "child-directed" apps requires stricter oversight. Currently, the "Kids" or "Educational" label appears to function primarily as a content descriptor rather than a privacy assurance. We argue that app stores and educational procurement frameworks (such as ST4S) should mandate that any application marketing itself to children must meet a verified technical baseline, specifically, the absence of advertising and behavioural analytics SDKs before being granted a "Kids" classification.

Second, regulatory frameworks must explicitly prohibit "idle telemetry." Our data shows that 89.3% of apps transmit data before user interaction. Policy interventions should require that non-essential SDK initialisation be technically deferred until affirmative consent is obtained. This would align technical architecture with the legal principle that consent must be prior and informed.

The privacy policies currently require on average university-level literacy, which indicates a move toward standardised, accessible disclosure formats. Rather than expecting parents to parse complex legal texts, regulators should enforce the use of plain summaries capped at a secondary-school reading level. These measures would shift the burden of safety from parents back to the developers and platforms profiting from the educational data economy.

VI. Conclusion

This study provides an empirical examination of the privacy and security risks and the transparency of educational Android applications that are commonly used by Australian schools or recommended by relevant authorities. We performed static and dynamic analysis, where the static analysis revealed the extensive use of embedded tracking and analytics libraries, and the dynamic analysis demonstrated that many applications commence network activity before meaningful user interaction, and routinely transmit persistent identifiers, as well as rich device profiles to multiple endpoints. We then analysed the privacy policy of the applications, and discovered that the privacy disclosures are often incomplete or misleading. The policy texts overall scored unsatisfactorily in our readabilty tests, with the scores indicating that an average parent would struggle to comprehend the privacy policies.

Our corpus of applications is focused on Australia and Android; therefore, the results might differ for iOS and other platforms, or other countries and regions. Applications evolve fast, and our measurements could only capture specific versions and time windows. For the dynamic analysis, the traffic was recorded on test devices configured for interception with a man-in-the-middle. We used a rooted emulator, with a user-installed certificate to bypass the SSL pinning of the applications. This may alter SDK behaviour as some SDKs might detect emulators or rooted devices, and some traffic flows may be under-observed. Our domain-to-recipient mapping might be inconcise, as domain fronting/CDNs and multi-user cloud services might blur the boundaries. Some privacy policies were excluded due to being inaccessible, and the readability scores are approximations that might be influenced by other factors such as the presense of non-English text, meaning the scores may not perfectly reflect comprehensibility especially for non-English speakers.

Future work can expand in several directions. First, platform and region coverage can be broadened by conducting a similar analysis on iOS applications and comparing the results with non-Australian app markets to distinguish global trends from regional patterns. Second, a longitudinal update analysis would enable researchers to track how changes in SDKs, Google Play

policies, and evolving privacy legislation influence developer behaviours over time. Finally, a categorical analysis comparing free and paid applications could help determine whether these groups differ in their behaviour patterns or in the degree to which their practices align with stated privacy policies.

ACKNOWLEDGEMENTS

We would like to thank UNSW Human Rights Institute for their funding to support this research.

REFERENCES

[1] L. Pangrazio and A. Bunn, "Assessing the privacy of digital products in australian schools: Protecting the digital rights of children and young people," *Computers & Education Open*, vol. 6, p. 100187, 2024.

[2] NSW Department of Education, "Digital devices and online services for students (policy)," https://education.nsw.gov.au/policy-library/policies/pd-2020-0471, 2025, department policy setting expectations for device and online services use in NSW public schools. Accessed 31-Oct-2025.

[3] Baulkham Hills North Public School, "Byodd recommended apps," https://baulkhamn-p.schools.nsw.gov.au/learning-at-our-school/byodd/byoddrecommendedapps.html, 2025, example NSW public-school iPad app list for BYODD classes. Accessed 31-Oct-2025.

[4] Lindfield Public School, "Stage 3 byod apps list 2023 [pdf]," https://lindfield-p.schools.nsw.gov.au/content/dam/doe/sws/schools/l/lindfield-p/notes/2023/notes/Stage_3_BYOD_Apps_List_2023.pdf, 2023, example NSW Stage 3 BYOD required/recommended apps list. Accessed 31-Oct-2025.

[5] L. P. School, "Bring your own device student agreement and responsibilities 2023 [pdf]," https://lindfield-p.schools.nsw.gov.au/content/dam/doe/sws/schools/l/lindfield-p/notes/2023/notes/2023_Stage_3_LPS_BYOD_Student_User_Agreement.pdf, 2023, bYOD program agreement showing school-level device program practice. Accessed 31-Oct-2025.

[6] Victorian Department of Education, "Software for victorian schools | arc," https://arc.educationapps.vic.gov.au/software, 2025, overview of the Arc platform for Victorian government schools. Accessed 31-Oct-2025.

[7] V. D. of Education, "Software for victorian schools | arc — software catalogue," https://arc.educationapps.vic.gov.au/software/catalogue, 2025, central catalogue of department-provided classroom software. Accessed 31-Oct-2025.

[8] I. Reyes, P. Wijesekera, J. Reardon, A. E. B. On, A. Razaghpanah, N. Vallina-Rodriguez, and S. Egelman, ""won't somebody think of the children?" examining coppa compliance at scale," *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 3, pp. 63–83, 2018.

[9] R. Sun, M. Xue, G. Tyson, S. Wang, S. Camtepe, and S. Nepal, "Not seen, not heard in the digital world! measuring privacy practices in children's apps," in *Proceedings of the ACM Web Conference 2023 (WWW '23)*, 2023, pp. 2166–2177.

[10] R. Carlsson, S. Rauti, S. Laato, T. Heino, and V. Leppänen, "Privacy in popular children's mobile applications: A network traffic analysis," in *2023 46th International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE, 2023.

[11] OAIC, "Notifiable data breaches report: January-june 2024," *Notifiable Data Breaches Report*, 2024. [Online]. Available: https://www.oaic.gov.au/__data/assets/pdf_file/0013/242050/Notifiable-data-breaches-report-January-to-June-2024.pdf

[12] N. D. of Education, "Notifying individuals affected by july's cyber incident," 2021. [Online]. Available: https://education.nsw.gov.au/news/latest-news/notifying-individuals-affected-by-julys-cyber-incident

[13] H. R. Watch, "Online learning products enabled surveillance of children," 2022. [Online]. Available: https://www.hrw.org/news/2022/07/12/online-learning-products-enabled-surveillance-children

[14] [Online]. Available: https://www.k12dive.com/news/parents-educators-received-explicit-image-in-seesaw-hacking/631951/

[15] N. Alomar, J. Reardon, A. Girish, N. Vallina-Rodriguez, and S. Egelman, "The effect of platform policies on app privacy compliance: A study of child-directed apps," *Proceedings on Privacy Enhancing Technologies*, vol. 2025, no. 3, pp. 170–191, 2025.

[16] J. Pimienta, J. Brandt, T. Bethe, R. Holz, A. Continella, L. Jibb, and Q. Grundy, "Mobile apps and children's privacy: a traffic analysis of data sharing practices among children's mobile ios apps," *Archives of Disease in Childhood*, vol. 108, no. 11, pp. 943–945, 2023.

[17] "Gdpr," https://gdpr-info.eu/.

[18] D. G. Krutka, R. M. Smits, and T. A. Willhelm, "Correction to: Don't be evil: Should we use google in schools?" *TechTrends*, vol. 65, p. 432, 2021.

[19] S. Zuboff, "The age of surveillance capitalism," in *Social theory re-wired*. Routledge, 2023, pp. 203–213.

[20] F. Ciclosi, G. Varni, and F. Massacci, "Gdpr in the small: A field study of privacy and security challenges in schools," in *2025 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2025, pp. 1197–1214.

[21] V. Zhong, S. McGregor, and R. Greenstadt, ""i'm going to trust this until it burns me" parents' privacy concerns and delegation of trust in k-8 educational technology," in *32nd USENIX Security Symposium (USENIX Security 23)*. Anaheim, CA: USENIX Association, 2023, pp. 5073–5090. [Online]. Available: https://www.usenix.org/conference/usenixsecurity23/presentation/zhong

[22] Safer Technologies 4 Schools (ST4S). (2025) Safer technologies 4 schools. [Online]. Available: https://st4s.edu.au/

[23] Ministry of Education (New Zealand). (2025, Jun.) St4s quick guide: Safer technologies for schools. NZ Ministry of Education. [Online]. Available: https://web-assets.education.govt.nz/s3fs-public/2025-06/1222%20ST4S%20-%20Quick%20Guide%20to%20Safer%20Technologies%20for%20Schools_AW.pdf

[24] Safer Technologies 4 Schools (ST4S). (2025) St4s vendor guide: Assessment process and criteria. [Online]. Available: https://st4s.edu.au/st4s-vendor-guide/

[25] eSafety Commissioner. (2019) Safety by design: Principles. [Online]. Available: https://www.esafety.gov.au/sites/default/files/2019-10/SBD%20-%20%20Principles.pdf

[26] eSafety Commissioner. (2025) esafety toolkit for schools. [Online]. Available: https://www.esafety.gov.au/educators/toolkit-schools

[27] Australian Government Department of Education. (2024) Australian framework for generative ai in schools. Originally released 2023; updated 2024. [Online]. Available: https://www.education.gov.au/schooling/resources/australian-framework-generative-artificial-intelligence-ai-schools

[28] Information Commissioner's Office (ICO). (2024) Age-appropriate design code (children's code). First published 2020; guidance updated. [Online]. Available: https://ico.org.uk/for-organisations/uk-gdpr-guidance-and-resources/childrens-information/childrens-code-guidance-and-resources/age-appropriate-design-a-code-of-practice-for-online-services/

[29] UK Department for Education. (2025) Cyber security standards for schools and colleges. Ongoing guidance, first published 2021. [Online]. Available: https://www.gov.uk/guidance/meeting-digital-and-technology-standards-in-schools-and-colleges/cyber-security-standards-for-schools-and-colleges

[30] F. T. Commission. (2025) Children's online privacy protection rule (coppa), 16 cfr part 312. [Online]. Available: https://www.ecfr.gov/current/title-16/chapter-I/subchapter-C/part-312

[31] ——. (2020) Complying with coppa: Frequently asked questions. [Online]. Available: https://www.ftc.gov/business-guidance/resources/complying-coppa-frequently-asked-questions

[32] F. of Privacy Forum and SIIA. (2024) Student privacy pledge. [Online]. Available: https://studentprivacypledge.org/home/

[33] C. Sense. (2025) Common sense privacy program. [Online]. Available: https://privacy.commonsense.org/

[34] Ministry of Education, Culture, Sports, Science and Technology (MEXT). (2025) Guideline on education information security policy. [Online]. Available: https://www.mext.go.jp/a_menu/shotou/zyouhou/detail/1397369.htm

[35] G. Wang, J. Zhao, M. Van Kleek, and N. Shadbolt, "Protection or punishment? relating the design space of parental control apps and perceptions about them to support parenting for online safety," *Proc. ACM Hum.-Comput. Interact.*, vol. 5, no. CSCW2, 2021.

[36] Z. Zhang, G. Wang, J. Zhao, M. Van Kleek, and N. Shadbolt, "Who is the user? the "identity crisis" of children's apps," in *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)*. ACM, 2024.

[37] M. Stoilova, S. Livingstone, and R. Nandagiri, "Digital by default: Children's capacity to understand and manage online data and privacy," *Media and Communication*, vol. 8, no. 4, pp. 197–207, 2020.

[38] L. Desimpelaere, L. Hudders, and D. Van de Sompel, "Children's and parents' perceptions of online commercial data practices: A qualitative study," *Media and Communication*, vol. 8, no. 4, pp. 163–174, 2020.

[39] P. C. Kumar, M. Subramaniam, J. Vitak, T. L. Clegg, and M. Chetty, "Strengthening children's privacy literacy through contextual integrity," *Media and Communication*, vol. 8, no. 4, pp. 175–184, 2020.

[40] O. Williams, Y.-Y. Choong, and K. Buchanan, "Youth understandings of online privacy and security: A dyadic study of children and their parents," in *Proceedings of the Nineteenth Symposium on Usable Privacy and Security (SOUPS 2023)*. Anaheim, CA, USA: USENIX Association, 2023, pp. 399–416. [Online]. Available: https://www.usenix.org/system/files/soups2023-williams.pdf

[41] L. Jibb, E. Amoako, M. Heisey, L. Ren, and Q. Grundy, "Data handling practices and commercial features of apps related to children: A scoping review of content analyses," *Archives of Disease in Childhood*, vol. 107, no. 7, pp. 665–673, 2022.

[42] I. Milkaite, R. De Wolf, E. Lievens, T. De Leyn, and M. Martens, "Children's reflections on privacy and the protection of their personal data: A child-centric approach to data protection information formats," *Children and Youth Services Review*, vol. 129, p. 106170, 2021.

[43] A. Ekambaranathan, J. Zhao, and G. Chalhoub, "Navigating the data avalanche: Towards supporting developers in developing privacy-friendly children's apps," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 7, no. 2, June 2023.

[44] [Online]. Available: https://drive.google.com/drive/folders/1LpRZPbTHHMrPqN67KrdRhGUDeNVF3I9b?usp=drive_link

[45] [Online]. Available: https://github.com/MobSF/Mobile-Security-Framework-MobSF

[46] O. Figueira, R. Trimananda, A. Markopoulou, and S. Jordan, "Diffaudit: Auditing privacy practices of online services for children and adolescents," in *Proceedings of the 2024 Internet Measurement Conference (IMC '24)*. ACM, 2024.

[47] Google, "Manage installations (firebase installations / fid)," https://firebase.google.com/docs/projects/manage-installations, 2025, describes Firebase Installation IDs used by Firebase services. Accessed 31-Oct-2025.

[48] Amplitude, "How amplitude identifies your users," https://amplitude.com/docs/get-started/identify-users, 2025, device ID / Amplitude ID / User ID model. Accessed 31-Oct-2025.

[49] R. Aghili, H. Li, and F. Khomh, "An empirical study of sensitive information in logs," *arXiv:2409.11313*, 2024, evidence of PII leakage in software logs. Accessed 31-Oct-2025. [Online]. Available: https://arxiv.org/abs/2409.11313

[50] Sentry, "Data scrubbing and pii controls," https://docs.sentry.io/security-legal-pii/scrubbing/, 2025, includes SDK/server-side scrubbing and `send-default-pii`. Accessed 31-Oct-2025.

[51] Google, "Best practices to avoid sending personally identifiable information (pii)," https://support.google.com/analytics/answer/6366371, 2025, mandate to avoid sending PII to Google. Accessed 31-Oct-2025.

[52] Google, "Google play families policies & families self-certified ads sdk program," https://support.google.com/googleplay/android-developer/answer/9893335, 2025, requires self-certified ad SDKs when serving ads to children. Accessed 31-Oct-2025.

[53] R. Binns, U. Lyngs, M. Van Kleek, J. Zhao, T. Libert, and N. Shadbolt, "Third party tracking in the mobile ecosystem," *arXiv preprint arXiv:1804.03603*, 2018. [Online]. Available: https://arxiv.org/abs/1804.03603

[54] M. Backes, S. Bugiel, and E. 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 (CCS)*, 2016, pp. 356–367, shows third-party libraries can add vulnerabilities or misuse inherited permissions. [Online]. Available: https://trust.cispa.saarland/publication/derr-16-ccs/derr-16-ccs.pdf

[55] B. Silva, D. Denipitiyage, S. Seneviratne, A. Mahanti, and A. Seneviratne, "Entailment-driven privacy policy classification with llms," in *2024 Conference on Building a Secure & Empowered Cyberspace (BuildSEC)*. IEEE, 2024, pp. 8–15.