Measuring and Engineering Privacy Protections

Nikita Samarin

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-231
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-231.html

December 20, 2024

Measuring and Engineering Privacy Protections

by

Nikita Samarin

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Doctor Serge Egelman, Co-chair
Professor David Wagner, Co-chair
Associate Professor Raluca Ada Popa
Professor of Law in Residence Chris Jay Hoofnagle

Fall 2024

Measuring and Engineering Privacy Protections

Abstract

Measuring and Engineering Privacy Protections

by

Nikita Samarin

Doctor of Philosophy in Computer Science

University of California, Berkeley

Doctor Serge Egelman, Co-chair

Professor David Wagner, Co-chair

Modern software often fails to meet privacy regulations and user expectations due to evolving legal landscapes, complex requirements, and a lack of structured engineering processes. Despite robust principles like Privacy by Design and formal regulatory frameworks such as the GDPR and CCPA, developers struggle to translate abstract obligations into actionable technical requirements. The resulting breaches and data misuse erode user trust and highlight systemic failures in embedding privacy within software systems.

This dissertation explores the causes of these failures and proposes professionalizing privacy engineering as a solution. Through an in-depth review of current regulations, two empirical studies of privacy engineering failures, and interviews with practicing privacy engineers, it identifies the persistent challenges developers face. These include unclear guidance for operationalizing legal mandates, inadequate technical tools, poor organizational incentives, and limited expertise in navigating complex privacy frameworks.

The research argues that formalized privacy engineering roles—encompassing technical, legal, and ethical understanding—can help ensure privacy requirements are integrated from the earliest stages of software development. By defining specialized skill sets, establishing clearer processes, and creating measurable metrics for success, privacy engineering professionals can bridge the gap between lofty data protection principles and practical implementation in code.

Ultimately, this dissertation concludes that professionalizing privacy engineering encourages a proactive, systematic, and ethically grounded approach to privacy. It calls on organizations, regulators, and policymakers to support these roles, offer clearer technical guidance, and align incentives so that privacy evolves into an intrinsic product quality rather than a compliance afterthought. This shift stands to improve user trust, reduce costly breaches, and better align digital technologies with fundamental rights and societal values.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This dissertation addresses the systemic failures of developers in building software that adheres to privacy regulations and policies, emphasizing the challenges they face amid an evolving landscape of privacy risks and complex regulatory environments. The evolving landscape of privacy requirements poses substantial challenges for organizations, software developers, and other key stakeholders in meeting their privacy obligations. These challenges contribute to frequent privacy breaches and data misuse, undermining public confidence in digital systems. This thesis argues that the professionalization of privacy engineering can mitigate these failures by equipping specialists to navigate intricate privacy requirements and effectively embed privacy into software systems. The research in this thesis demonstrates how formal privacy engineering techniques could have prevented these issues by conducting two empirical studies on privacy engineering failures. Ultimately, the dissertation aims to show how professionalized privacy engineering can address consistent failures in the creation, implementation, and verification of software privacy requirements.

The goal of this dissertation is to address the systemic failures of developers in building software that adheres to privacy regulations and policies. But what are these regulations and policies, what role do developers play in meeting these obligations, and why is it so challenging to engineer software that respects these privacy requirements?

1.1 Evolving landscape of privacy risks

The widespread adoption of software applications across various domains and platforms—ranging from mobile devices to cloud-based services—has led to unprecedented levels of data generation and processing [103]. Everyday actions such as browsing social media, using mobile apps, or shopping online contribute to a continuous stream of personal data being collected, often without explicit user consent or even awareness [136].

Against this backdrop, personal data has emerged as a highly valuable commodity in the digital economy, underpinning the business model of many software companies [236]. Developers and software providers leverage this data to gain insights into consumer behavior, personalize services, and drive targeted advertising. For instance, the digital advertising industry heavily relies on user data to deliver personalized ads, a market projected to increase from $628.8 billion in 2022 to $1.2 trillion by 2027 [23].

Software companies have long emphasized the benefits that consumers receive in exchange for their personal information. For instance, organizations describe the monetization of personal information as a means of providing free or discounted software to their users [41]. Others argue that the collection and processing of personal data increases the value of digital services by allowing them to meet individual consumer needs in a way that is easier, more efficient, or more enjoyable than it would be otherwise [199].

This thesis does not explore the *benefits* that sharing personal information brings to consumers. Instead, it focuses on the *costs* of this exchange, namely, the erosion of consumer privacy and the increased risk of harm that follows, not only to individuals but also to communities and societies.

Privacy concerns are not new, nor are they limited to the era of mass communications. In 1890, American lawyers Warren and Brandeis famously proposed the *right to privacy* as an extension of legal protections to include the individual's right to be let alone, safeguarding personal thoughts, emotions, and private life from unwarranted public disclosure [218]. Different cultures have also long recognized the distinction between *public* and *private* spheres of life with historical roots in ancient Greek philosophical discussions [188].

Although the concept of *privacy* is not novel, the privacy risks and the potential for harm created by modern information technologies certainly are, resulting from the widespread adoption of these technologies and their ability to communicate with each other. High-profile incidents affecting millions of individuals, such as the Equifax data breach [235] or Facebook-Cambridge Analytica scandal [102], revealed not only the scale of modern privacy risks but their evolution from individual harms (e.g., identity theft) to societal-level harms (e.g., influencing voter behavior during a democratic election process or amplifying existing social inequities). These and many other privacy incidents, alongside the increased visibility of harms, have amplified public demand for greater transparency, control, and accountability in how personal information is collected and used by private and public entities.

These mounting concerns, coupled with the recognition that existing laws are inadequate at regulating modern information technologies, prompted governments and regulatory bodies worldwide to enact privacy and data protection[1] rules aimed at safeguarding individual privacy rights and addressing the costs associated with the personal data economy. Notable examples include the European Union's General Data Protection Regulation (GDPR) [68] and the California Consumer Privacy Act (CCPA) [123], which impose strict requirements

[1]This thesis uses the terms *privacy* and *data protection* interchangeably while recognizing that these terms may have differing meanings depending on the context or jurisdiction.

on data collection, processing, consent, and user rights. These laws and, more importantly, their underlying privacy and data protection principles form the landscape of the privacy rules and public policies governing the information practices of modern software applications.

1.2 Developers struggle to meet privacy obligations

The passage of privacy regulations, such as the GDPR and CCPA, was hailed as a monumental milestone in consumer data protection, increasing organizational costs of not adhering to the now-codified data protection principles [8]. These costs represent an assortment of regulatory penalties, including fines, legal injunctions, and consent decrees, in addition to other costs, including the loss of trust and damage to a company's reputation and brand image.

Despite the laudable goals of these laws and their notable achievements, they have nonetheless translated to a complex and fragmented landscape of privacy regulations [20]. Legislation and policies across different jurisdictions embody varying conceptualizations of privacy, creating a multifaceted regulatory environment where what is permissible in one context may be forbidden or restricted in another.

In addition to the convoluted landscape of privacy requirements, prior research has identified many factors that contribute to developers' failures to account for privacy requirements in software applications. These factors include, but are not limited to:

- limited knowledge of privacy principles, frameworks, and laws [170, 93, 157];
- lack of usable tools and frameworks for privacy implementation [104, 171, 190];
- lack of organizational culture around privacy [184, 17, 93];
- misplaced responsibility for privacy implementation [9, 153, 24]; and
- misaligned incentives around privacy and other business functions [64, 124, 189].

The consistent failure of developers to engineer software that respects privacy requirements necessitates the need for solutions that address these underlying factors. Recognizing that no single work can address all of these factors, this thesis moves the needle in that direction by measuring privacy engineering lapses and investigating how the formalization of privacy engineering can address the systemic factors contributing to these failures.

1.3 Engineering privacy

There is a longstanding recognition among the scholarship that the ability to successfully implement and validate privacy requirements in software systems (in other words, *engineer privacy*) depends on the ability of those building the systems (i.e, software developers) to

understand and navigate the complex landscape of privacy requirements, as formed by various standards, industry codes of conduct, consumer concerns, best practices, professional spheres of influence, regulatory environments, legal agreements, among many other requirement sources [182, 89, 90]. While much of the existing work has focused on making it easy for the end user to make decisions about their privacy, we now also observe the rise of research focusing on the developers as the key stakeholders in protecting privacy. This line of research attempts to identify the gaps that prevent developers from navigating the privacy requirements landscape and provide usable tools to facilitate compliance with these requirements.

This thesis argues that such developments are not new. Over the last several decades, researchers have observed the evolution of security engineering from a footnote in the broad software engineering curriculum to a significantly more mature and specialized field with experts trained in information, systems, networks, and other types of computer security. We can witness similar trends in the field of safety and trust engineering, which is additionally shaped by disciplines outside of computing and engineering. Similar to the challenge of engineering *security* or, arguably more so, *safety* or *trust*, engineering *privacy* requires operationalizing an essentially contested concept [139].

Notwithstanding the enormous benefit of teaching software engineers the foundational knowledge about these ideas and offering them the path of "least resistance" in the form of usable tools and concrete software design patterns, there is also something to be gained by acknowledging the inherent complexity of these concepts that require specialized knowledge and even distinct career paths and professional development that might not be representative of the average software engineer. This thesis underscores the challenges of software developers in meeting their privacy obligations while emphasizing how specialists in privacy engineering (i.e., *privacy engineers*) can help software teams identify and mitigate privacy risks.

The value of privacy engineers in the business world is directly tied to their ability to navigate the complex and often ambiguous landscape of privacy requirements. As specialists, they bring a methodical approach to identifying, interpreting, and implementing privacy measures in software development and data management processes. Their expertise becomes increasingly valuable as organizations grapple with evolving privacy regulations, heightened public awareness, and the potential reputational and financial risks associated with privacy breaches.

Privacy engineers ensure data utilization aligns with privacy rights and expectations in the current data-driven business environment. Their work not only helps organizations maintain compliance with a diverse array of global privacy laws but also contributes to building trust with customers and stakeholders. As privacy concerns continue to influence public discourse and consumer behavior, the role of privacy engineers in reconciling technical capabilities with privacy expectations becomes increasingly central to the success and sustainability of modern businesses.

This dissertation argues that the professionalization of the privacy engineering field offers

answers to some of these challenges. Using two concrete examples when privacy engineering fails, leading to leakage of personal data and the inability to exercise privacy rights, this dissertation aims to demonstrate that these examples could have been mitigated if formal privacy engineering techniques were employed. As such, it aims to answer the research question of how professionalized privacy engineering can address the consistent failures in the creation, implementation, and verification of privacy requirements in software.

1.4 Research overview

The remainder of this dissertation is structured to build a comprehensive understanding of why developers fail to embed privacy into their software, and to propose professionalized privacy engineering as a necessary corrective measure.

Following the introduction, the next portion of the thesis (**Chapter 2**) establishes the conceptual and legal groundwork for modern privacy protection, tracing how historical principles and frameworks like Fair Information Practice Principles (FIPPs), OECD Guidelines, and Privacy by Design (PbD) evolved. This chapter then links these foundational concepts to the Software Development Lifecycle (SDLC), illustrating how integrating privacy from the outset can prevent costly retrofits later.

Building on this foundation, the subsequent **Chapter 3** presents methods to empirically measure existing privacy protections in software. It reviews techniques for examining declared and actual data practices, including the use of runtime analysis, code inspection, and automated tools, which set the stage for the two empirical studies of privacy engineering failures that follow.

The first measurement study (**Chapter 4**) explores how developers implement privacy rights, focusing on the California Consumer Privacy Act (CCPA) and its "right to know" provisions. By comparing what developers disclose in privacy policies and verifiable consumer request responses against their apps' actual data transmissions, it identifies pervasive non-compliance and uncovers systemic barriers preventing developers from meeting legal mandates.

Next, our second empirical investigation (**Chapter 5**) examines how developers implement privacy protections in secure messaging apps. Analyzing these apps' use of Google's Firebase Cloud Messaging (FCM), it reveals unintended leakage of message data to third-party infrastructure. Even privacy-focused apps, the study shows, are not immune to design oversights and ambiguous requirements.

Shifting from these measurement studies, **Chapter 6** discusses the challenges that developers face when implementing privacy requirements and makes the case for a specialized professional (a *privacy engineer*) tasked with translating the privacy protection principles into practical implementations.

The following **Chapter 7** draws on interviews with practicing privacy engineers. Their

first-hand accounts highlight the complexities of translating legal and ethical principles into technical requirements, underscore the importance of organizational support, and point to the urgent need for professionalized privacy engineering roles.

Finally, the conclusion (**Chapter 8**) synthesizes these insights, arguing that professionalizing privacy engineering can address these persistent failures. It summarizes key contributions, discusses implications for industry and policymakers, and outlines avenues for future research.

Chapter 2

Foundations of Modern Privacy Protection

This chapter provides a foundational understanding of privacy protection principles and how they have evolved into frameworks and design approaches that shape modern software development. Building on early legal scholarship and concepts like "the right to be let alone," the chapter traces the development of Fair Information Practice Principles (FIPPs), their internationalization through the OECD Guidelines, and the subsequent emergence of Privacy by Design (PbD) as a proactive, systems-level integration of privacy safeguards.

The chapter then examines how these principles have influenced regulatory frameworks worldwide, contrasting the European Union's comprehensive General Data Protection Regulation (GDPR) with the United States' fragmented sectoral approach supplemented by state-level initiatives such as the California Consumer Privacy Act (CCPA). By situating privacy principles and legal requirements within the context of the Software Development Lifecycle (SDLC), it underscores the importance of incorporating privacy from the earliest phases of software engineering. This sets the stage for exploring the practical challenges that developers face when integrating privacy considerations into their systems.

The purpose of the research in this dissertation is to improve the integration of privacy protection principles into software applications through privacy engineering methods. But how do we measure the efficiency of these methods? In other words, how do we know whether we are successful in engineering privacy principles into software products? To answer this question, we first need to understand the principles that underlie modern privacy protection and how these principles relate to software engineering.

This chapter provides the necessary background about the fundamental privacy protection principles and their codification in modern privacy laws and regulations. It then positions these principles within the context of modern software development practices.

2.1 Protecting what exactly?

Everyone seems to agree that *privacy* is important and deserves protection. But what does *privacy* actually mean? This turns out to be a rather challenging question to answer, as the concept of privacy resists easy definition and is contested across multiple disciplines and cultural contexts. Similarly to other essentially contested concepts such as *art* or *freedom*, attempts to pin down a single, universal definition have repeatedly run up against the myriad ways in which personal, social, economic, and political factors shape what privacy means to different groups at different times [139].

Privacy is thus far from a static or universal concept; rather, its contours change depending on the specific circumstances under which information is shared and used. Nissenbaum's theory of contextual integrity, for example, highlights that the appropriateness of sharing information depends on the norms that govern particular contexts [142]. Under this view, consent to share personal data for one purpose (e.g., "to receive medical treatment") cannot be viewed as consent to share the same information for another unrelated purpose (e.g., "to receive targeted advertising"), as different informational norms would govern these different contexts.

Some legal scholars, including Solove and Hartzog, have also emphasized the importance of embracing a pluralistic and dynamic understanding of privacy to address the complex challenges of modern information practices [97]. Instead of chasing an elusive singular definition of privacy (e.g., "control over personal information"), they underscored the need for a more nuanced and multifaceted approach to privacy regulation that focuses on protecting against harms instead of viewing privacy through a particular lens [178].

This thesis focuses on principles of *privacy protection* rather than *privacy* itself. By talking about privacy protection, we shift the discussion to methods and processes instead of ways to conceptualize privacy itself.

2.2 The right to be let alone

The conceptual and practical approaches to privacy protection have undergone significant changes since their earliest formulations in the late nineteenth century. Early legal scholarship provided the intellectual foundation for understanding privacy as a legal right and social value, which later informed the development of robust regulatory frameworks and strategic principles guiding the design and implementation of privacy protections.

The seminal work of Warren and Brandeis famously defined privacy as "the right to be let alone." In their influential 1890 Harvard Law Review article, they argued that the legal system's recognition of a right to privacy was both necessary and justified in response to the emergent mass media technologies of their time [218]. This idea set the stage for courts and legislatures to consider privacy as a standalone interest, distinct from defamation or property claims, and central to the evolving notion of civil liberties. Subsequent scholar-

ship expanded on the premise that privacy was central to human dignity, autonomy, and democratic values [76].

By the mid-twentieth century, perspectives on privacy began to shift from static notions of intrusion to dynamic concerns regarding data flows and information management. Westin's conceptualization of privacy as the *right to control* personal information positioned privacy at the intersection of emerging computer technologies and data processing [220]. This framing anticipated the increased importance of how personal data could be collected, stored, analyzed, and disseminated. These considerations would become more consequential with the widespread adoption of networked computing and digital communication systems.

2.3 Fair Information Practice Principles (FIPPs)

The shift from legal discourse to structured privacy protection principles emerged as computing and data processing became widespread. In the early 1970s, increased concern over the capacity of information technology to store, process, and analyze personal data led to significant policy developments in privacy protection. Among these developments was the 1973 U.S. Department of Health, Education, and Welfare (HEW) report [19] that introduced normative standards for processing personal information that would later become known as the Fair Information Practice Principles (FIPPs).

The FIPPs correspond to a set of key values, including transparency, consent, and accountability, that are integral to the processing of personal data and maintaining public trust in data-intensive systems [77]. They provided a conceptual framework for shaping legislation in the United States, such as the U.S. Privacy Act of 1974 [51], and influenced the data protection laws of numerous other countries. Over time, the FIPPs evolved and were integrated into both sector-specific and general privacy legislation, maintaining their importance in evaluating the adequacy of privacy protections [180].

2.4 OECD Guidelines

The rapid globalization of commerce and communications technologies in the late twentieth century pressed for a more harmonized approach to data protection principles. Recognizing the international dimension of data flows, the OECD established its Guidelines on the Protection of Privacy and Transborder Flows of Personal Data in 1980 [61]. These Guidelines, updated over time to reflect evolving technological landscapes, represent a milestone in codifying a set of internationally accepted privacy principles [25].

The OECD Guidelines expanded FIPPs into eight core principles that balanced the interests of data protection with the legitimate needs of international business and government cooperation:

- **Collection Limitation**: Personal data should be collected by lawful and fair means, with knowledge or consent of the data subject, and be limited to what is necessary.
- **Data Quality**: Personal data should be accurate, complete, up-to-date, and relevant to the purposes for which they are to be used.
- **Purpose Specification**: The purposes for which personal data are collected should be specified at the time of collection, prior to any processing.
- **Use Limitation**: Personal data should not be used for purposes other than those specified at collection except with consent or legal authority.
- **Security Safeguards**: Personal data should be protected by reasonable security safeguards against risks such as unauthorized access, destruction, use, modification, or disclosure.
- **Openness**: Organizations should maintain transparency about their practices and policies regarding personal data processing.
- **Individual Participation**: Individuals should have the right to obtain confirmation of data holdings, receive communications about their data, and challenge data for rectification or erasure.
- **Accountability**: Data controllers should be responsible for complying with measures that give effect to these principles.

The OECD Guidelines, revised in 2013 to address the challenges posed by emerging technologies and big data analytics, have influenced data protection statutes and privacy laws worldwide, including the European Union's 2016 General Data Protection Regulation (GDPR) [68] and the 2018 California Consumer Privacy Act (CCPA) [123]. The information principles in these Guidelines set a baseline from which data governance policies could be compared, ensuring minimal standards across countries with varying legal traditions [143].

2.5 Privacy by Design (PbD) framework

While FIPPs and the OECD Guidelines offered substantive guidance on the handling of personal data, they did not intrinsically mandate embedding privacy protections into technological systems. The concept of Privacy by Design (PbD) emerged in the 1990s to fill this gap by advocating that privacy considerations be integrated throughout the entire system engineering process [40].

Developed by Ann Cavoukian, the former Information and Privacy Commissioner of Ontario, Canada, PbD promotes the incorporation of privacy considerations into the design and operation of IT systems, infrastructures, and business practices from the outset, rather than as an afterthought [39]. PbD consists of the following seven foundational principles:

- **Proactive not Reactive; Preventative not Remedial**: Privacy risks should be anticipated and prevented before they materialize, rather than waiting for privacy breaches to occur.
- **Privacy as the Default Setting**: Personal data should be automatically protected in any system or business practice, with no action required by the individual.
- **Privacy Embedded into Design**: Privacy should be integrated into the architecture and design of systems and business practices, not bolted on as an add-on after the fact.
- **Full Functionality – Positive-Sum, not Zero-Sum**: Privacy should be accommodated without degrading system functionality, demonstrating that it's possible to have both privacy and security.
- **End-to-End Security – Full Lifecycle Protection**: Privacy protection should extend securely throughout the entire lifecycle of the data involved, from collection to destruction.
- **Visibility and Transparency**: All component parts and operations of business practices and technologies should remain visible and transparent to both users and providers.
- **Respect for User Privacy**: Architects and operators must keep the interests of the individual paramount by offering strong privacy defaults, appropriate notice, and user-friendly options.

While PbD marked significant progress in operationalizing privacy protection, its implementation has revealed persistent challenges. Organizations often struggle to translate these principles into technical and organizational measures, particularly in complex systems where privacy requirements compete with other design goals or business objectives [181]. For example, implementing "privacy as the default setting" requires careful consideration of what constitutes appropriate defaults across different contexts and user groups. Similarly, "privacy embedded into design" demands sophisticated architectural decisions that may not be obvious to development teams.

Despite these challenges, PbD offers us a framework to describe the privacy-friendliness of different software systems. By examining how software systems do not adhere to these principles, we can begin to form the narrative around the challenges developers face when engineering privacy and propose concrete solutions.

2.6 Regulatory implementation

With an established set of principles and the emergent philosophy of PbD, privacy regimes began to coalesce into robust and enforceable regulatory frameworks. In Europe, the General Data Protection Regulation (GDPR), which came into effect in 2018, represents the culmination of decades of regulatory evolution originating from the 1995 Data Protection

Directive [67]. The GDPR operationalizes FIPPs, OECD principles, and PbD within a comprehensive, extraterritorial legal framework that imposes strict obligations on data controllers and processors, mandates data protection impact assessments, and significantly enhances the rights of data subjects [68]. GDPR's broad scope and substantial penalties have positioned it as a global benchmark, influencing the development of privacy laws in Latin America, Africa, and Asia [86].

In contrast, the United States has historically followed a more sectoral and fragmented approach, relying on a combination of federal, industry-specific, and state-level regulations [169]. At the federal level, the Federal Trade Commission (FTC), through its enforcement of Section 5 of the FTC Act [48], relies on a "notice-and-choice" model that emphasizes transparency and consent but does not impose comprehensive data protection standards across all industries [179]. Instead, specific statutes address particular data categories: the Health Insurance Portability and Accountability Act (HIPAA) for health information [50], the Children's Online Privacy Protection Act (COPPA) for children's data [47], and the Gramm-Leach-Bliley Act (GLBA) for financial data [49]. In recent years, states have taken the initiative to fill the regulatory gaps. The California Consumer Privacy Act (CCPA) [123], effective since 2020 and amended by the California Privacy Rights Act (CPRA) [215], introduces broader consumer rights—such as the right to access, delete, and opt out of data sales—moving U.S. privacy protections closer to the comprehensive approach seen in the EU.

The divergence between US and EU approaches reflects different philosophical foundations: the US treats privacy primarily as a consumer protection issue, while the EU considers it a fundamental right [25, 116]. Despite these differences, both approaches draw from the same foundational principles, demonstrating the enduring influence of early privacy frameworks on modern privacy engineering practice.

2.7 Engineering privacy

Building upon the conceptual and regulatory foundations discussed in the preceding sections, integrating privacy protection principles into software engineering practice emerges as a critical step toward ensuring data protection throughout the entire system lifecycle. While frameworks such as the FIPPs, the OECD Guidelines, and the PbD paradigm establish high-level norms and guidelines, their effective realization depends on translating these principles into concrete engineering methodologies, tools, and development processes [57]. The engineering of privacy-preserving systems, therefore, necessitates a systematic approach that aligns normative principles with established processes for developing software [88].

This thesis models software engineering activities throughout the entire system lifecycle using the Software Development Lifecycle (SDLC), which represents a series of phases for engineering software, ranging from requirements elicitation and architectural design to implementation, verification, and maintenance [156]. The fundamental goal of SDLC is to provide

a systematic approach that ensures that the final software product meets the stakeholders' expectations[1], is delivered on time, is cost-effective, and adheres to quality standards.

Although organizations employ various methodologies, a typical SDLC encompasses seven fundamental phases:

1. **Planning**: Defining project scope, objectives, and initial requirements;
2. **Analysis**: Gathering detailed requirements and analyzing software needs;
3. **Design**: Creating the software architecture and detailed specifications;
4. **Implementation**: Writing code and building the software;
5. **Testing**: Verifying functionality and compliance with requirements;
6. **Deployment**: Releasing the software to production; and
7. **Maintenance**: Ongoing support, updates, and modifications.

Modern development approaches such as Agile or DevOps can compress or iterate through these phases more rapidly, but fundamental activities remain consistent. Each phase presents distinct opportunities and challenges for privacy engineering. We particularly emphasize the importance of *requirements engineering*: a subset of SDLC activities that focus on understanding the needs and constraints of all stakeholders—end users, customers, developers, regulators, and others—and translating these needs into clear, measurable, and testable specifications [130].

Adopting principles such as "Privacy by Design" involves identifying privacy requirements from the outset and embedding them into the system's architecture, policies, and data handling practices [89]. By explicitly capturing privacy requirements early in the SDLC, software teams can ensure compliance with relevant regulations (e.g., GDPR in the EU, sectoral and state-level laws in the U.S.), adhere to established privacy frameworks (like the FIPPs), and ultimately safeguard user data throughout the system's lifecycle [37]. This proactive approach to integrating privacy requirements reduces the risk of costly retrofits, improves user trust, and aligns the resulting software with global standards for personal data protection.

The next chapter surveys related work on the challenges that developers face when integrating privacy protection principles into the design and implementation of their software applications. In doing so, we set the stage for exploring two real-world examples of failures to account for these principles and understanding how privacy engineering methods address the factors underlying these privacy lapses.

[1]The stakeholders of a software product can be varied and include not only the software company and its customers but also regulators, policymakers, and society at large.

Chapter 3

Measuring Existing Privacy Protections

This section explores how researchers can identify privacy-threatening software behavior and compare it against expected practices. It frames the successful implementation of privacy principles as the integration of corresponding privacy requirements into software. By focusing on privacy-by-design principles, regulatory mandates, and contractual obligations, the approach outlined here emphasizes the translation of abstract principles into concrete, testable requirements. Success is measured by examining the actual information practices of the software, such as data collection, sharing, and processing, and assessing whether they align with the expected practices dictated by these privacy requirements. Existing techniques, including code analysis, runtime behavior observation, and privacy policy examination, provide insights into developers' adherence to privacy standards. Overall, the work surveyed highlights methodological advancements that aid in identifying discrepancies and informs future efforts to guide developers toward more effective privacy protection measures in their applications.

After reviewing the fundamentals of modern privacy protection principles and their connection to the software development process, this chapter surveys related work on ways to identify privacy-threatening software behavior and compare it against expected practices.

3.1 Privacy protection and information practices

Understanding why software developers fail to engineer and maintain privacy protections successfully requires us to recognize the inherent difficulty in measuring *success* in the first place. How do we know whether we have successfully 'baked' privacy into our software application? What does implementing privacy principles even mean in the first place?

As mentioned in the previous chapter, this thesis views the implementation of privacy principles as engineering privacy requirements identified from those principles. In other

words, we view successful privacy implementation as capturing the relevant privacy needs of different stakeholders, translating those needs into concrete software requirements, implementing them in software design and code, verifying their correct implementation, and continually changing the requirements as needs evolve. To scope this problem further, the research in this dissertation focuses on privacy-by-design principles, their regulatory implementation, and, to a lesser extent, contractual obligations[1] as sources of privacy requirements. We can then frame the success of implementing privacy protection principles in terms of whether the software meets or complies with the privacy requirements collected from these sources.

We are still left with several challenges. For one, translating abstract principles into software requirements is a process that is open to wide interpretation. For instance, the *respect for user privacy* principle can be expressed in a wide number of requirements, such as ensuring that appropriate consent is collected or privacy-friendly default options are used. Similarly, the implementation of these requirements will also depend on the software in question. Implementing user consent might be straightforward on a personal computer, but what about voice assistants and other IoT devices without a visual input-output interface? How should we approach the implementation of these privacy requirements, given this varying context?

To address these challenges, the research in this thesis focuses on privacy requirements concerning the *information practices* of a given software application, such as the collection, sharing, or processing of personal data. Therefore, we do not consider privacy requirements that pertain, for instance, to avoiding dark patterns or other manipulative design more generally. Moreover, we investigate the implementation of privacy requirements within the context of mobile apps, given the prevalence of this type of software and its ability to collect and process a wide range of personal information, such as personal identifiers, geolocation, health-related data, and inferred behavioral profiles [21].

Given this framing of the problem, we can measure how well a developer implements privacy protection in their application by identifying its information practices and then comparing them against the practices that we would expect to observe if the application was fully compliant with its privacy requirements. While it might not be easy or even possible to identify what the expected information practices are for every possible software-requirement pair, we can evaluate successful privacy protection for many meaningful privacy requirements. For instance, the requirement to stop the collection of some user identifier under certain conditions (e.g., after the user clicks the opt-out button) means that the expected information practice is to *not* collect this identifier.

The main challenge remains around measuring the existing software information practices and comparing them against expected practices in meaningful ways. One possible approach

[1]Contractual obligations in this context refer to the information practices that a software company outlines to their users in their legal documents, such as their terms of service or a privacy policy. An example might be "this company does not collect or share your personal information" or "our software employs reasonable security measures that protect your personal data."

to identify existing information practices is to ask the app developers. Although surveys and interviews can shed some light on these practices, these methods rely on developers to accurately represent how their application handles personal information. But modern software applications are complex—and even with developers' best efforts, they may be unable to enumerate all the possible ways in which personal information gets acquired, processed, and shared within their apps [170].

Instead, the research presented in Chapters 4 and 5 relies on methods that directly analyze the software product (e.g., website, mobile app, IoT device, etc.) and its artifacts (e.g., underlying code, run-time behavior) to establish its information practices. This approach allows us to identify the 'ground truth' and compare it against the expected practice for a given privacy requirement. In what follows, we provide an overview of prior work that focused on developing the methods and tools used to analyze software applications before discussing ways in which the scholarship applied these techniques to identify privacy-threatening information practices across different contexts.

3.2 Identifying privacy-relevant information practices

Numerous studies have also investigated the security and privacy ramifications of mobile apps (e.g., [72, 110, 197, 191]). Most current methods for evaluating mobile app information practices depend on static analysis [113, 78, 85, 234], which examines the app's source code without executing it. However, this technique is limited as it can only identify the potential behaviors of a program, not if and to what degree the program exhibits them. For instance, it is generally infeasible to predict the full set of execution branches that a program will take. Alternative methods, such as taint tracking [66], which tracks the flow of data as it propagates through the application, come with their own challenges, including affecting app stability [36].

A newer approach involves adding instrumentation to the Android operating system to monitor apps' access to personal information at runtime [222, 223, 202, 224]. This allows researchers to investigate different app behaviors, including app-associated network traffic. Prior solutions to monitoring mobile app transmissions generally involve using proxy software (e.g., Charles Proxy,[2] mitmproxy,[3] etc.) and suffer from serious shortcomings. First, they route all the device traffic through the proxy, without automatically attributing traffic to a specific app running on the device. While some traffic may contain clues (e.g., content and headers that may identify apps, e.g., HTTP `User-Agent` headers), other traffic does not, and attributing traffic to the app is a laborious and uncertain process [160]. Second, proxies often cannot automatically decode various obfuscations, including TLS with certificate pinning. Instead, by capturing traffic from the monitored device's OS, these issues are eliminated. This approach can bypass certificate pinning, extract decryption keys from memory, and

[2] https://www.charlesproxy.com/
[3] https://mitmproxy.org/

map individual sockets to process names, thereby offering precise attribution to specific apps. We use this approach in Chapters 4 and 5 to measure the privacy-relevant information practices of mobile apps at scale.

Analysis of Privacy Disclosures

In some cases, we also want to identify the *declared* information practices in addition to the *observed* ones. For example, many privacy laws that operationalize the *visibility and transparency* principle do so by requiring software developers to offer privacy policies and other notices that inform users about the software's information practices. In effect, these become self-reported presentations of information practices, akin to asking the software developer to describe them. Nevertheless, analyzing privacy policies and other disclosures made by businesses allows us to capture another perspective on how software providers present their information practices.

Prior research has focused on understanding apps' and websites' declared information practices by analyzing disclosures made in privacy policies [96, 12, 216, 233, 234]. Some proposed systems, such as POLICHECK [12], MAPS [233] and HPDROID [69], which automate the process of comparing disclosures made in privacy policies about how user data is used, collected, or shared with personal data transmissions observed as a result of performing technical analyses [216, 12, 233, 234, 175]. The literature also proposed systems, such as Polisis [96], PI-Extract [32] and PrivacyFlash [232], which made it possible to transform privacy policies into formats that are more understandable to users or auto-generate policies that reflect actual app behaviors.

Additionally, Google's Play Store requires developers to provide privacy labels [83]. Privacy labels communicate information practices to users in a visually succinct way. For example, apps may list the data types (e.g., names, phone numbers, identifiers) collected and shared with third parties. As with privacy policies, these privacy labels are required by the Google Play Store's terms of service to be thorough and complete [83]. However, Google states in their guidelines that "transferring user data to a 'service provider'" should not be disclosed as data sharing in the app's privacy labels [83], limiting their scope and potential utility. Other studies have also demonstrated the inconsistencies between privacy labels and privacy policies [187], privacy labels in the Google Play Store and Apple App Store for the same apps [164], and practices disclosed in privacy labels and behaviors observed among iOS apps [114, 229].

3.3 Comparing practices against requirements

After identifying the actual and declared information practices using these approaches, we can attempt to compare them against expected practices in meaningful ways. Defining expected information practices is not always trivial and may not always be possible (e.g., if there are incompatible or unfeasible privacy requirements). Prior research has, therefore,

also worked on defining expected information practices depending on the specific privacy requirement and measured how much the observed practices of real-world software match those expectations.

For example, academic scholarship has applied these methods to investigate potential privacy violations in online systems, including mobile apps and websites [141, 163, 94]. To examine the extent to which apps comply with privacy regulations, researchers relied on static and dynamic app analysis tools to identify potential legal violations at scale [163, 94, 70, 141, 162, 106]. These studies identified a range of deceptive data collection and transmission practices and highlighted the need for stronger enforcement actions by regulators.

Linden et al. [126] found that disclosures made in privacy policies improved as a result of GDPR enforcement (which requires the implementation of PbD principles), but that more improvements would have to be made before they can be considered usable and transparent to users. Other recent studies have also examined the accuracy of disclosures made in privacy policies [11, 145, 216].

In summary, we examined how comparing actual and declared information practices against those mandated by privacy requirements and policies can serve as a way to identify ways in which developers fail to engineer privacy protection. Researchers, policymakers, and other stakeholders can use these methods to measure the landscape of privacy protection in different contexts and requirements. These insights can then be used to drive change in privacy protection through better translations of privacy requirements and better software design and implementation. The next two chapters present studies that build on these methods to identify ways in which developers fail to implement privacy protection principles into their apps, while underscoring privacy implications for end users and ways to improve the alignment between the expected and actual information practices.

Chapter 4

How do App Developers Implement Privacy Rights?

The California Consumer Privacy Act (CCPA) provides California residents with a range of enhanced privacy protections and rights. Our research investigated the extent to which Android app developers comply with the provisions of the CCPA that require them to provide consumers with accurate privacy notices and respond to "verifiable consumer requests" (VCRs) by disclosing personal information that they have collected, used, or shared about consumers for a business or commercial purpose. We compared the actual network traffic of 109 apps that we believe must comply with the CCPA to the data that apps state they collect in their privacy policies and the data contained in responses to "right to know" requests that we submitted to the app's developers. Of the 69 app developers who substantively replied to our requests, all but one provided specific pieces of personal data (as opposed to only categorical information). However, a significant percentage of apps collected information that was not disclosed, including *identifiers* (55 apps, 80%), *geolocation data* (21 apps, 30%), and *sensory data* (18 apps, 26%) among other categories. We discuss improvements to the CCPA that could help app developers comply with "right to know" requests and other related regulations.

In the previous chapter, we discussed ways to establish privacy-relevant software behavior and compare it with expected information practices. By performing this comparison, we can measure the gap between the current and expected implementation of privacy requirements. This chapter[1] applies this framework to measure the failure of Android app developers to comply with the privacy requirements defined by the underlying data protection principles of *transparency* and *access*.

[1]This technical chapter is based on work previously published in a peer-reviewed journal [166].

4.1 Introduction

On January 1, 2020, the California Consumer Privacy Act (CCPA) went into effect [123]. Modeled after the European Union's General Data Protection Regulation (GDPR) [68], the CCPA is designed to increase the control of California consumers over their personal information and offer stronger privacy protections than those available to data subjects in the rest of the United States. Among other provisions, the CCPA requires certain companies operating in California to disclose their data collection and sharing practices and respond to consumers' requests to access their personal information held by the company. This "right to know" allows individuals to obtain information that belongs to them and confirm that businesses comply with the data practices stated in their privacy notices.

The required notice of data practices and the right to know what personal information was collected by a business embody two crucial principles of data protection: individual participation and openness [62]. Businesses comply with these principles by posting privacy policies and responding to "subject access requests" (SARs) from consumers (known as "verifiable consumer requests" or "VCRs" under the CCPA). Although these principles appear in other privacy frameworks, regulations such as the GDPR and the CCPA define a stricter set of requirements and impose heavier penalties for non-compliance than previous data privacy regimes. For instance, the CCPA prescribes what businesses need to include in their privacy notices and how they should respond to VCRs.

When implemented correctly, the "right to know" can greatly benefit consumers. First, accurate information about data collection and sharing practices is necessary to allow consumers to make informed decisions about whether and what information to disclose to the business or whether to seek alternatives, if necessary. Second, the ability to request data pertaining to oneself allows consumers to amend inaccurate information held by the business (the right to rectification) or transmit information to another business of their choosing (the right to data portability). Awareness of the information held by the business can also prompt consumers to request data relating to them be deleted (the right to erasure) [111] and lead to the adoption of other privacy-enhancing technologies (PETs). As such, the right to know and other privacy rights enabled by it serve to advance consumers' informational self-determination and increase their bargaining power in digital environments.

Unfortunately, scholarship has already identified shortcomings of other privacy rights granted by the CCPA. For instance, Consumer Reports found that consumers struggled to opt out of the sale of their personal information and were at least *"somewhat dissatisfied"* with the processes they had to go through 52% of the time [132]. More recently, Nortwick and Wilson [212] found that many websites required to comply with CCPA either failed to provide users with options to request not selling their data to third parties or provided options that suffered from major usability issues. Other studies have also found issues with similar privacy laws enacted earlier, most notably the GDPR in Europe, including evidence of non-compliance by app developers [233] and personal information leakage by abusing the right of access [60]. These shortcomings have to be addressed to ensure that the regulations'

stated goal of furthering privacy protections for consumers is adequately fulfilled.

Although prior studies have focused on the impacts of the CCPA and the GDPR [213, 60, 59, 120], we were unable to find any empirical studies measuring the compliance of businesses with the "right to know" requirements set by the CCPA, specifically in the context of mobile applications ("apps"). We thus pose the following research question: **To what extent do Android app developers comply with the provisions of the CCPA that require them to maintain accurate privacy notices and respond to consumers' access requests by disclosing personal information that they have collected about them?** We focus on mobile apps in large part because they present inherent and unique privacy risks, as the devices they are installed on accompany their users throughout their everyday lives and provide access to a wide range of sensitive information, including geolocation, health, and biometric data.

We examined the data practices of 160 top-ranked Android mobile app developers from the U.S. Google Play Store, who we expected to meet the definition of a "business" regulated under the CCPA and, thus, be required to comply with its provisions. Due to ethical concerns, we focused only on the subset that publicly posted information indicating they would be responsive to users' CCPA requests. We then submitted VCRs to these 109 companies by following the CCPA-specific disclosures in their privacy policies, and compared their responses with the actual data practices that we identified through static and dynamic analysis of their mobile apps. We found that at least 39% of the apps shared device-specific identifiers and at least 26% shared geolocation information with third parties without disclosing it in response to our requests. Furthermore, of the 69 app developers who substantially responded to our requests, all but one disclosed the specific pieces of collected personal information, but only 36% included the CCPA-required categories of third-party data recipients in their responses.

The results of our work hold several important policy implications. We argue that regulators—and, in particular, the California Privacy Protection Agency (CPPA)—should issue more guidance for developers to help them better comply with the CCPA and its latest amendment, the California Privacy Rights Act (CPRA). Such guidance should include examples of personal information that can be collected from consumers' mobile devices and emphasize the legal obligations for developers who meet the definition of a "business" regulated by CCPA. One such obligation is to provide accurate responses to consumers' VCRs; regulators should remind developers that they have to provide all of the requested information, including the categories of personal information and third parties, and ensure that the provided categories are specific to the consumer in question.

4.2 Background and related work

We provide an overview of the CCPA, including information about the required notices and disclosures to consumers. We then highlight prior work that investigated the accuracy

of disclosures made in privacy policies, the efficacy of subject access request mechanisms, and the potential privacy violations that exist in online systems, including mobile apps and web-based systems.

Overview of CCPA's Requirements

The California Consumer Privacy Act (CCPA) is a state statute that was signed into law in June 2018, becoming effective on January 1, 2020 and enforceable on July 1 of the same year [35]. The CCPA secures a number of privacy rights for California consumers and imposes new obligations on companies operating in California. In contrast to the EU's General Data Protection Regulation (GDPR) [68], the CCPA only applies to for-profit businesses that do business in California and meet any of the following conditions [123]:

- Have a gross annual revenue of over $25 million;
- Buy, receive, or sell the personal information of 50,000 or more California residents, households, or devices; or
- Derive 50% or more of their annual revenue from selling California residents' personal information.

Importantly, the CCPA grants consumers the *right to be notified* about the data collection and sharing practices of a business and, after such collection has taken place, the *right to know* the personal information that the business has pertaining to them.

Notices to Consumers. The CCPA requires businesses to provide consumers with a *privacy policy* and a *notice at collection*. The purpose of the privacy policy is "to provide consumers with a comprehensive description of a business's online and offline practices regarding the collection, use, disclosure, and sale of personal information and of the rights of consumers regarding their personal information" [35]. The CCPA regulations require that the privacy policy is "posted online through a conspicuous link using the word 'privacy' [...] on the download or landing page of a mobile application" and include the following information [35]:

- Explanation that a consumer has the right to request that the business disclose what personal information it collects, uses, discloses, and sells;
- Instructions for submitting a verifiable consumer request;
- Description of the process for verifying the consumer request, including information the consumer must provide;
- Categories of personal information the business has collected about consumers in the preceding 12 months;

- Categories of personal information, if any, that the business has disclosed or sold in the preceding 12 months and, for each category, the categories of third parties with whom the information was shared;
- Categories of sources from which the personal information is collected; and
- Business or commercial purpose for collecting or selling personal information.

In addition to the privacy policy, the CCPA requires businesses to provide consumers "with timely notice, at or before the point of collection, about the categories of personal information to be collected from them and the [collection] purposes" in the form of a notice at collection [35]. Although businesses might choose to maintain a separate notice at collection, they can also provide a link to the section of the privacy policy containing the required information, as long as the company presents the link at or before the collection of personal information [35].

Verifiable Consumer Requests. The CCPA grants another fundamental privacy right to California consumers, namely, the right to know the personal information that a business has collected pertaining to them. Consumers can exercise this right by submitting a "verifiable consumer request" (VCR). The CCPA requires businesses to provide two or more designated methods for submitting VCRs. Furthermore, businesses have 10 days to confirm the receipt of the VCR and 45 days to complete the request, either by providing the requested data or denying it. The CCPA allows businesses to extend the timeline by up to an additional 45 days, provided they inform the requester of the extension and its reasons.

As part of the VCR, consumers can request the same types of information that is required to be in a privacy policy (see list above). However, unlike the general data practices described in the privacy policy, the response to the VCR has to be specific to the consumer making the request. Crucially, in addition to the aforementioned information, a consumer can also request that the business disclose *specific* pieces of personal information that it has collected about the consumer. Unlike the GDPR [154], the CCPA does not require companies to disclose specific names of third parties with whom they share the consumer's personal information.

The CCPA regulations describe the steps that businesses must take to verify the identity of the consumer submitting the VCR. Such verification is crucial to ensure that the company does not disclose a consumer's personal information to an unauthorized party. Simultaneously, businesses need to carefully consider the type and sensitivity of personal information to ensure that their verification procedures do not prevent consumers from successfully exercising their privacy rights. Furthermore, a business should avoid collecting additional personal information solely for the purposes of identity verification (unless absolutely necessary), it cannot impose fees for verification, and should implement reasonable security measures to prevent unauthorized disclosure of consumers' personal information. If a business maintains a password-protected account with the consumer, they can employ that existing account's

authentication mechanisms to verify the consumer's identity. Otherwise, the business is required to verify the requester's identity to a "reasonable degree of certainty" by matching either two (before disclosing categories of personal information) or three (before disclosing specific pieces of personal information) data points provided by the consumer with data points maintained by the business.

The CCPA defines a consumer as a California resident "however identified, including by any unique identifier,"[2] which means that consumers need not use their real names to identify themselves when making VCRs. That is, the CCPA allows consumers to use pseudonyms when transacting with businesses and exercising their privacy rights, and does not require that they divulge their legal names to make VCRs (i.e., for verification, it only needs to match the personal information previously collected by the business).

Comparison with the GDPR

The EU's General Data Protection Regulation (GDPR), which went into effect on May 25, 2018, is considered to be one of the most comprehensive data protection laws to date [33]. Similar to the CCPA, the GDPR offers strong privacy protections to individuals and imposes obligations on businesses conducting business in Europe. In particular, the GDPR also requires companies to disclose their data collection and sharing practices in a privacy policy and respect individuals' right to be informed and right of access to personal information pertaining to them.

Despite the similarities in the rationale between the CCPA and GDPR, there are also important differences with regard to the scope and application of specific provisions [105, 58]:

1. **Personal Scope.** The GDPR applies broadly to entities that establish the means and purposes of the processing of Europeans' personal information, covering natural and legal persons, for-profit, non-profit, and public entities, small and large organizations, irrespective of their size or revenue. On the other hand, the CCPA only applies to for-profit businesses subject to the criteria enumerated in Section 2.1.
2. **Material Scope.** The CCPA excludes specific categories of personal information from its scope of application covered by industry-specific federal privacy laws, whereas the GDPR does not feature such exceptions. For instance, medical information covered by the Health Insurance Portability and Accountability Act (HIPAA) and financial information covered by the Gramm-Leach-Bliley (GLB) Act are both outside of the scope of application of the CCPA.
3. **Required Notices.** The CCPA requires covered businesses to disclose in their privacy policies the categories of personal information collected, sold, or disclosed for a business purpose in the preceding 12 months.

[2] Cal. Civil Code §1798.140(g).

4. **Right of Access.** The CCPA mandates that companies provide personal information requested by the consumer under the right to know "in a portable and, to the extent technically feasible, readily usable format that allows the consumer to transmit this information to another entity without hindrance," effectively establishing the right to data portability. In contrast, the GDPR separates the right of access and the right to data portability, which have their own conditions.
5. **Procedures.** The CCPA requires organizations to respond to consumers' request in 45 days starting with the receipt of the request, extendable once by an additional 45 days. The GDPR requires covered entities to respond within one month, extendable once by an additional two months.
6. **Penalties.** The GDPR empowers competent data protection authorities to both assess any violations of the law and directly issue fines to entities. In contrast, the Attorney General of the State of California is responsible for assessing violations of the CCPA and bringing civil actions against the offending businesses to seek statutory damages in court.

The next section provides an overview of prior studies investigating the efficacy of the right of access, primarily under the GDPR. We believe that although the methodologies and general findings are applicable to our study, the highlighted differences between the two data protection laws also necessitate the present exploration of businesses' compliance with the CCPA.

Efficacy of Subject Access Requests

Our work relates to prior studies that investigated how effective subject access request (SAR) mechanisms are in helping data subjects exercise their rights [213, 60, 59, 120, 4, 7, 31, 210]. In [209], SARs were sent to 38 third-party businesses in an effort to evaluate how they comply with Article 15 of the GDPR, and the study showed that most failed to properly disclose all relevant user data in their responses to the requests. Urban et al. [210] sent SARs to 36 organizations and found that 58% delayed responding to the requests. Kröger et al. [120] sent similar requests to app developers over a period of a few years and identified potential weaknesses in the processes developers followed to handle and respond to such requests, which continued to exist even after GDPR became enforceable. Similarly, the results of sending SARs to businesses in [18] highlighted the difficulty they experienced finding all data needed to respond to the requests. The authors also emphasized the importance of using automation whenever possible when responding to SARs and developing templates that businesses can follow so that they can reach a state of "legal certainty," where they can be assured that they are in compliance with laws that provide users with the right to access their data. Tolsdorf et al. [198] identified data incompleteness and inconsistency issues when evaluating the accuracy of information displayed in privacy dashboards for a number of service providers.

Study	Request Count	Response Count	GDPR or CCPA?	Policy Analysis?	App Analysis?
This	**109**	**80 (73%)**	**CCPA**	**Yes**	**Yes**
[4]	109	62 (57%)	GDPR	No	No
[210]	36	32 (89%)	GDPR	Yes	No
[34]	14	14 (100%)	GDPR	No	No
[31]	326	212 (65%)	GDPR	Yes	No
[151]	150	112 (75%)	GDPR	No	No
[120]	225	43–58 (19–26%)	GDPR	No	No
[18]	60	44 (73%)	—	Yes	No
[59]	40	34 (85%)	GDPR	No	No
[60]	55	51 (93%)	GDPR	No	No
[209]	38	16 (42%)	GDPR	Yes	No
[99]	150 apps 120 sites	43%	GDPR	No	No

Table 4.1: Comparison of key metrics with related work.

Herrmann and Lindemann [99] observed that businesses were more likely to respond to data deletion requests than subject access requests, and identified websites that adopted SAR mechanisms that made them vulnerable to revealing their users' data in their responses to adversarial data access requests. In a number of other studies, researchers further examined how businesses' SAR mechanisms can be used by adversaries to extract subjects' personal data through social engineering attacks (e.g., impersonation) [60, 59, 151, 52, 28, 34]. Di Martino et al. [60] showed how these types of attacks can be mounted against a number of organizations by relying on information that is available to the public. In their follow-up work [59], they proposed alternative approaches to authenticating data subjects that can help businesses strengthen their SAR mechanisms by reducing the likelihood of leaking subjects' personal data when responding to data access requests made by adversaries. Jordan et al. [108] focused specifically on addressing the problem of how organizations can respond to data access requests that do not have corresponding user accounts.

While prior work has investigated organizations' responses to SARs from a number of different perspectives, we believe that the literature is yet to paint a complete picture on the extent to which responses to SAR are consistent with disclosures made in privacy policies and actual system behaviors. Researchers investigated whether SAR processes are sufficiently explained in privacy policies or aligned with the requirements of applicable laws and compared privacy policy disclosures to responses to SARs [209, 28, 31, 29, 210], but we are unaware of studies that compared organizations' responses to actual system behaviors. We systematically compare information obtained from the three sources of information we considered: privacy policies, responses to SARs and actual app behaviors.

Researchers also studied the usability of subject access request and deletion mechanisms from a number of different angles, including the ease of initiating the requests as well as the extent to which the content of the responses can be understood by average users [213, 219, 92, 29, 208]. After investigating users' awareness of their rights under the GDPR in [121], researchers found that users do not have sufficient understanding of their "right to data portability." Habib et al. [92] uncovered challenges users experience with locating information related to how to exercise their privacy rights and correctly using the privacy controls made available to them by businesses. Veys et al. [213] observed how real users interacted with the content of the responses obtained from businesses after requesting to download their data. They found that most responses are yet to be considered *accessible* to users and identified areas where future improvements can be made to better align these responses with user expectations [213]. Urban et al. [208] highlighted the importance of improving the designs of current user-facing tools provided by organizations to allow users to understand how their data is used. After studying the extent to which responses to SARs submitted to Twitter are empowering real users to understand how their data was used in ad targeting, Wei et al. [219] similarly found content-related issues that might negatively affect how understandable and readable ad explanations are to users. Table 4.1 compares some of the key metrics of this study with those of prior work.

4.3 Methodology

We aim to uncover contradictions between personal information...

1. that we record being collected and transmitted by an app using dynamic and static analysis;
2. disclosed to us in response to a "right to know" request we made after using the mobile app; and
3. that the app developer claimed to collect in their app's privacy policy.

The following sections cover each part of the study in more detail. We additionally describe our procedure for selecting the Android apps that we examined, as well as our procedure for testing the apps and submitting the verifiable consumer requests.

Dataset

We focused on the 8 top-ranked Android mobile apps in the 20 Google Play Store categories that have the highest number of cumulative app installs. Companies developing these apps fall or can be reasonably inferred to fall under the CCPA definition of a "business."[3] We

[3] A "business" includes mobile app developers that are for-profit entities and conduct business in California (i.e., make their applications available in California) and meet at least one of the following three

selected only one mobile app (with the highest user install count) per developer in order to have the ability to match the personal information disclosed by the developer with the app that we tested and to examine a broader range of developer practices for responding to VCRs.

Furthermore, we replaced certain apps that we were unable to test. This included apps, for instance, that required business accounts, financial information, or additional hardware devices. This selection procedure produced a total of 160 unique apps, which we downloaded with their privacy policies in November 2021.

It is important to note that, although our procedure was designed to select developers that we expected to be covered by the CCPA, the resulting list was only an approximation (i.e., we could not be sure that all of these developers were *actually* subject to the CCPA), as we used the number of app installs to gauge the total number of California consumers from whom an app may have collected personal information. Nonetheless, we could not be sure, and as an ethical matter, we did not want to waste people's time by submitting VCRs to organizations that were not required to respond to them. Thus, we further limited our study to only those companies that explicitly mentioned CCPA in their privacy policies. Under the FTC Act[4] (and various other state consumer protection laws), businesses in the U.S. are prohibited from materially misrepresenting their practices to consumers. This includes making false statements in privacy policies, which the FTC enforces (e.g., [46]). Thus, any business that states in their privacy policy that they respond to CCPA VCRs must actually do so, regardless of whether or not they are *actually* covered by the CCPA.

Two researchers from our team independently read the text of 160 privacy policies to determine whether or not each contained references to the CCPA. For cases without a majority consensus, a third researcher provided the tie-breaking vote. Our analysis indicated that out of the selected 160 apps, 109 (68%) include CCPA-specific disclosures in their privacy policies (with Krippendorff's alpha = 0.81, indicating an acceptable level of inter-rater agreement [118]). For the remainder of this paper, our discussion will focus primarily on these 109 apps.

App Analysis

We used an instrumented version of Android 9.0 (Pie) that monitored resource accesses (e.g., access to Android APIs) and logged all network traffic, regardless of the use of TLS. (Prior published work has applied a similar approach [161, 168, 12, 94, 9].) Because network traffic was captured at the OS level (as opposed to using a proxy), we were still able to observe and decrypt transmissions that were secured using certificate pinning. Since the values of identifiers (and other personal information) were known for each device, our tools

criteria: (1) collect the personal information of at least 50,000 consumers in California; (2) have an annual gross revenue in excess of $25 million, or (3) derives 50 percent or more of its annual revenues from selling California consumers' personal information (CCPA, 1798.140(c)).

[4] 15 U.S.C. §45.

automatically searched for various permutations in the captured network traffic, including hashes (e.g., MD5, SHA-1, SHA-256, etc.).

Using this instrumentation on Google Pixel 3a devices, we automatically recorded decrypted network traffic, which included destinations (i.e., hostname, port, IP) and payloads. Decrypted traffic payloads included API endpoints and key/value pairs. All network traffic was attributed to specific apps and their SDKs, using a combination of kernel-level instrumentation to attribute sockets to processes and stack inspection to identify specific SDKs. A variety of open-source tools for collecting network traffic can be used to verify our results and, we believe, reproduce our findings from scratch (e.g., [75, 148]). While the instrumentation was specifically written for Android Pie (9), which was released roughly three years prior to our testing, millions of people still use Pie (e.g., at the time that we conducted our study, roughly 20% of US Android users were using Pie or earlier [185]), many with CCPA rights. We also have no reason to believe that the same app binaries would be more/less compliant under newer Android versions.

Pseudonyms. Similar to [231], we generated pseudonyms and other fictitious values for different types of personal information covered by the CCPA to facilitate the subsequent search for this data in the logs produced by app testing and to improve the ecological validity of our study. Our motivation behind using "fake" data was to reduce the number of confounding variables: while all experimenters were California residents, if we used our real names and identifiers, we would not know whether data received from CCPA VCRs was collected by the company during the study period or before (or possibly from other sources).

The CCPA defines a consumer as a California resident "however identified, including by any unique identifier,"[5], therefore, the usage of fictitious data did not legally affect the requirement of the companies to respond to our requests. This provision ensures that companies that only collect pseudonyms are still subject to CCPA requests, while also disincentivizing companies from collecting additional personal information solely for the purpose of responding to requests. A physical address (and email, phone number, etc.) can be fictitious, so long as they can be used to identify the California consumer who is the data subject. Thus, the use of pseudonyms both reduced confounding factors and was legally valid.

We produced pseudonymous data using random value generators, such as the Random Lists [159] website and Faker Python package [107]. We obtained other types of personal information, including device identifiers and geolocation data, directly from our test devices. We present our data taxonomy in Appendix A.3, while Table A.3 provides examples of personal information that we used.

Testing Procedure. We manually tested the selected 109 apps, each for approximately 15-20 minutes using test phones with our instrumented version of the Android operating system. We set up each test phone—to be used by an individual tester in California—

[5]Cal. Civ. Code §1798.140(g).

to use its own set of pseudonymous identifiers, such as the phone number, email address, usernames, and other types of information. During each test, we created a user account for the app (if applicable) and input the predefined pseudonymous data corresponding to the specific test phone, as described above. We later searched for the predefined data values within the resulting test logs (which included captured network traffic), as well as performed an open-ended search to see if the app transmitted other personal data.

Data Recipients. Apps can transmit data both to first- and third-party destinations in order to deliver essential and non-essential functionality. Specifically, we might observe an app transmit the same personal information only to domains controlled by the app developer or to a combination of first- and third-party endpoints.

First, we categorized the observed destination domains as either first- or third-party for each tested app. Using the same approach as in [200], we tokenized the destination domain and the app package name. We then classified a specific domain as first party if its tokens appeared in the app's privacy policy URL or matched the package name's tokens, otherwise, we labeled the domain as third party. Next, we went over the resulting party labels for each domain and manually corrected any mistakes. For each third-party domain, we also obtained the effective second-level domains (eSLD) using `tldextract` and used it to locate the entity that controls it using Crunchbase, Netify, and other online resources. Two researchers from our team assigned a category to each third-party domain using the information that we obtained from our online search, which we then used to compare against the categories of recipients in VCR responses and privacy policy disclosures.

The CCPA recognizes that a first party can either directly or indirectly collect personal information.[6] As such, the collection of personal information via third parties (either service providers or third parties under the CCPA) still triggers the CCPA obligations on the first party as if the app developer directly collected the personal information itself. The liability of first parties for third-party app and website data collection has been affirmed by People of the State of California v. Sephora USA, Inc. [53].

For this reason, we labeled each data point (e.g., for purposes of Table 4.4) that we observed being captured and transmitted to a third-party domain (e.g., using SDKs, codebases, or other pieces of code in the app) as collected both by the first party (i.e., the app developer) and the third party. We categorized the data point as collected by the first party if the app transmitted it only to domain(s) controlled by the app developer.

Verifiable Consumer Requests

For each tested app, we identified directions in its privacy policy for how to submit a verifiable consumer request (VCR). To avoid abusing the time and resources of developers who do not have to comply with the provisions of the CCPA, we erred on the side of caution and only

[6] Cal. Civ. Code §1798.130(a)(3)(A).

submitted verifiable consumer requests to developers who explicitly referenced the CCPA in their privacy documents.

As part of each request, we asked to obtain all types of information that a business is required to provide under the CCPA in response to a consumer request:

1. specific pieces and categories of personal information requested, collected, and shared by the app;
2. categories of sources from which the personal information was collected;
3. business or commercial purposes for collecting the personal information; and
4. specific names and categories of third parties with whom the app developer shared personal information.[7]

We submitted each request from the same pseudonymous email account that was used to test the app. We employed email templates to ensure a level of uniformity when, for instance, we submitted the initial requests, sending follow-ups if the developer did not respond, asking for an alternative identity authentication mechanism, etc. We provide the email templates that we used to submit the requests and follow up with the developer in Appendix A.1. Nevertheless, some developers still instructed us to use an alternative method for submitting the request, such as a privacy management platform.

Privacy Policy Analysis

Additionally, we analyzed disclosures made in the privacy policies of tested apps using a deductive approach to qualitative coding. Our codebook contains codes for the collection and sharing of categories of personal information taken from Cal. Civil Code 1798.140. One of the authors with experience assisting companies in complying with the CCPA requirements developed the codes for the categories of third parties. We include the resulting codebook, code descriptions, and prompts in Appendix A.2.

As discussed previously, we first identified whether each policy contained references to the CCPA using the following prompt: "*Does this app developer include disclosures that reference the CCPA, either as part of the general privacy policy or as a standalone document?*" We then analyzed each of the 109 privacy policies containing CCPA-specific information to identify information about the developer's data collection and sharing practices. In particular, for each category of personal information defined under the CCPA (e.g., identifiers or geolocation), we examined whether an app developer collected or disclosed each category and to which category of recipients.

At least two annotators from our team first independently located the relevant privacy policies, and then used the prompts enumerated in Table 4.5 to locate the disclosures that

[7]CCPA does not require businesses to disclose specific third parties, however, some app developers opt in their privacy notices to provide that information upon request.

pertained to the collected and shared categories of personal information and the categories of third parties. We then computed Krippendorff's α to evaluate the inter-rater reliability on a per-question basis [119]. We resolved any divergences in our responses using a majority vote or, if a majority was absent, a third researcher independently provided the tie-breaking vote. After resolving the disagreements, we obtained a list of categories of personal information and recipients that we compared against our app analysis results.

Comparison

We compared these three data viewpoints to quantify the accuracy and completeness of the information disclosed by the developers. We first compared each specific piece of personal information that we observed being collected and shared with the specific pieces of information disclosed by the developer in the VCR, when applicable. In this case, we simply matched the values that we observed being collected and shared with the values provided to us by the developer. As mandated by the CCPA, we only accepted responses containing the values (and not just the types of information) to be valid with respect to disclosing the specific pieces of personal information.

Furthermore, we compared the categories of collected and shared information and the categories of recipients that we observed during app testing with the the same categories disclosed in the VCR and privacy policy. We only considered the categories disclosed in the VCR responses to be valid if we were able to sufficiently match them with the CCPA-defined categories of personal information and to our categories of third parties. These categories included common types of recipients that we observed across different app privacy policies and VCR responses, such as advertising networks, marketing partners, analytics providers, fraud and security, search engines, social media networks, payment processors, customer support providers, storage and infrastructure, affiliates, and law enforcement. We obtained the same categories of personal information and third parties from the privacy policies using the qualitative coding approach discussed previously. The CCPA-defined categories of personal information as well as the categorization of our own PII types are presented in Table A.3.

Once we had obtained these categories, we identified the categories that the developer had collected but not disclosed by looking at the difference between categories that we observed during app testing and the categories provided by the app developer in the privacy policy and VCR.

Ethics

We performed a study of institutional processes and did not collect data *about* individuals [100]. As such, our IRB determined that our study did not meet the legal definition of human subjects research, and therefore declined to review it. We nonetheless spent over a year deliberating how to conduct it ethically, including avoiding guessing whether a company was subject to CCPA, not incurring costs by asking legal questions, and making sure

correspondence was not perceived as legal threats, ethical issues that have come up for other researchers [134]. Instead, we performed a measurement study of publicly-available services by exercising our legal rights using the methods companies themselves prescribed.

We acknowledge that some companies may not have automated systems to process CCPA requests, and therefore processing our VCRs may have imposed costs on them. However, we believe that business' interests in this regard are outweighed by the public interest in understanding CCPA effectiveness. This is also a straw man argument: all individuals who made CCPA requests for our study were legitimately interested in learning about companies' privacy practices and made legally-valid requests to do so; that they additionally followed a prescribed methodology and shared the results for research purposes does not suddenly make the requests invalid or unethical. CCPA empowers California residents with *rights*, which must be honored regardless of intent.

4.4 Results

We present the results from submitting the VCRs, focusing on the methods available to do so, the types of information required to initiate and verify requests, and the percentage of developers who completed the requests, with an emphasis on the disclosure of the CCPA-specific information, as enumerated in Section 3.4. Furthermore, we compare the personal information provided to us by the developers with our dynamic analysis of their Android apps.

Access Requests

We analyzed the 109 apps with CCPA-specific information in their privacy policies. Whenever possible, we created an account with each app using an email address created specifically for this study and unique to the testing phone. As a result, we registered accounts with 91 (83%) apps.

The majority of developers (66%) provided at least two methods for submitting the VCR. The most common method was by email, with 71 (65%) companies offering it as an option. The next most common method was a dedicated VCR form or portal offered by 42 (39%) companies. Notably, 15 of these companies relied on OneTrust [201], a third-party suite of products that includes support for SAR management, with the remaining 27 either relying on another third-party provider or implementing their own solutions. We identified a number of other methods for submitting requests, including a phone number (25%), contact via customer support service (19%), physical mail (19%), account or in-app privacy settings (15%) or through a Google Form (2%).

Whenever possible, we submitted VCRs using email or a customer support service. In these cases, our messages to the companies included a self-attestation of California residence, as well as the pseudonyms and email addresses associated with the phone used for testing

Method	Count	Proportion
Email	71	0.65
Company DSAR Portal	27	0.25
Phone	27	0.25
Customer Support Service	21	0.19
Physical Mail	21	0.19
OneTrust DSAR Portal	15	0.14
Account Privacy Settings	11	0.10
In-App Privacy Settings	5	0.05
In-App Feedback Form	3	0.03
Google Form	2	0.02

Table 4.2: Distribution of methods for submitting VCRs.

the app. However, in 16 cases, the app developer directed us to use an alternative VCR submission method other than the one we had chosen. Ultimately, we submitted 52 VCRs via email and 6 VCRs via customer support or a feedback form out of the total 109 requests sent out. We submitted the remaining 51 requests either using a provided VCR portal (34%) or within an app's or account's privacy settings (13%).

When using a dedicated form, portal, or in-app privacy controls to submit the VCR, we generally received a confirmation of the request within the same user interface. For this reason, we focused on the 58 apps that required a free-form request submission to see if the companies would confirm the receipt of our request within the statutory 10 day period mandated by the CCPA. Out of these 58 companies, 40 (69%) explicitly confirmed our request, whereas the remaining 18 (31%) did not.

Companies also must verify the identity of consumers submitting VCRs to ensure they do not inadvertently disclose personal information to someone impersonating the data subject. Therefore, we also recorded information that we provided or any authentication steps we performed to verify our VCR (Table 4.3). We implicitly verified the ownership of our email address in 52 instances, when we made the request via email. For all other cases, 32 companies requested email verification after submitting the request, typically by clicking a link or providing a unique PIN sent to the testing email address. Furthermore, 35 companies required us to successfully log into our accounts either to submit or to verify the VCR.

App developers also requested specific pieces of personal information to match against their records, either as part of the initial VCR submission process or by following up with us after we submitted our requests. Most often, developers asked us to provide some basic information about ourselves, including, our email address (36 instances), full name (26), state (21), and country of residence (15). Developers also requested technical information that is not always easily accessible for smartphone users. In particular, we were asked to provide the Android Advertising ID (AAID) in 5 cases, a company-defined 'device' or 'user'

ID in 5 cases, and our current IP address in 2 cases. Table 4.2 presents a breakdown of the different types of information or actions required to verify the VCRs.

Some companies had more stringent requirements to complete their identity verification, either at the moment or after submitting the VCR. Five companies out of 109 required us to certify the accuracy of the provided information under penalty of perjury and 4 required a signed affidavit that, in at least one case, had to be notarized.[8] Furthermore, two companies requested proof of phone number ownership by providing a recent mobile operator bill, another two asked for photocopies of a government-issued ID, and one company outsourced identity verification to the ID.me service, which allows an individual to verify themselves either by providing a photocopy of their government-issued ID or their phone number to allow a look-up with the mobile operator records. Finally, one company asked us to "make [ourselves] available for a phone call with a [redacted] customer service representative who will call from [their] privacy line." In these instances when we could not furnish such documents, we requested an alternative verification method through logging into our account and providing details of that login to the company, if applicable. The CCPA regulations explicitly provide for such an alternative verification method for account-holders. Two companies agreed, and allowed us to verify our identity using the alternative verification method.

The majority of companies, namely 102 or 94%, did not ask for proof of our California residency. Out of the remaining 7 app developers, three asked us to provide proof of our address (e.g., a bank statement or a recent utility bill), one requested a government-issued ID showing California residency (e.g., a California driver's licence), one asked us to sign a declaration of California residency under the penalty of perjury, and the remaining two requested California state residency verification via ID.me and the phone call, as described previously.

Developer Responses

Out of the 109 requests that we sent out, we did not receive a response from the developer in 21 (19%) cases. In these instances, the developer either did not respond to the initial request or became unresponsive after a brief interaction, for instance, after asking for verification. In all of these cases, we followed up with the app developers at least once to confirm that they were unresponsive.

We were unable to verify our identity to the company's satisfaction in 5 (5%) other cases, as we were unable to produce the requested documentation and the company did not agree to use an alternative method. Finally, 3 (3%) developers could not verify our identity to a sufficient degree and, thus, did not respond with any personal information. We excluded these 29 cases from our analysis of the responses and focused on the remaining 80 responses.

[8] This is explicitly prohibited by regulations (§999.323(d)).

Table 4.3: Methods or information required to verify VCR.

Method or PII Type	Count
Email	36
Account Authentication	35
Email Authentication	32
Full Name	26
State of Residence	21
App-specific Information	18
Country of Residence	15
Username	9
Phone Number	7
Postal Address	6
Device or User ID	5
Android Advertising ID (AAID)	5
Certification w/ Penalty of Perjury	5
Signed Affidavit	4
Photocopy of a Government-issued ID	3
Phone Authentication	3
Current IP Address	2
Date of Birth	2
ID.me	1
Call with a Company Representative	1

Human vs. Automated Responses. We first identified the proportion of companies employing automation when responding to our VCRs. Similar to [209], we labeled responses that directly answered to our questions as "human." In contrast, we marked responses sent by a computer system (e.g., help desk ticketing software) or containing only generic privacy-related information as "automated." Out of 80 responses, we labeled 32 (40%) responses as "human" and the remaining 48 (60%) as "automated."

Follow-up Actions. We first examined the number of actions that the data subject would have to perform to successfully receive a response to their VCR, and the amount of time they would have to wait for the company to reply back. Across the 80 responses, we performed an average of 1.8 ($\pm$0.78, median = 1) actions to obtain our VCR response, including submitting the request, passing identity verification, following up with the developer, etc. The most actions that we performed was 4. Additionally, it took us 14.86 ($\pm$18.86, median = 7) days on average to receive responses to our VCRs, however, the average was skewed heavily by developers who instantly replied back with the response (e.g., if made through in-app account settings) and those that took extraordinarily long, with the longest duration to complete the request of 76 days.

Composition of the Response. Out of these 80 companies, 69 (63%) provided data in response to our request, 8 (7%) replied that they held no data on us and the remaining 3 (3%) told us to obtain the requested information directly from our account profile.

For the 69 companies that provided us data, we examined whether they provided all types that a business is required to provide under the CCPA (Section 3.4). All but one app developer provided us with specific pieces of information in their responses. However, compliance with other parts of the CCPA's right to know was less uniform. For instance, only 24 (35%) companies provided the categories of personal information collected from us, 18 (26%) provided the categories of personal information disclosed or sold to third parties, 25 (36%) provided the categories of those third parties, 30 (43%) responded with the business or commercial purpose for collecting or selling our personal information, and 23 (33%) disclosed the sources, from which our information was collected.

Compliance. The relatively high compliance with the request to provide specific pieces of information is not surprising, as many app developers are likely using tools to automatically respond to CCPA (and GDPR) requests by integrating with and pulling data from their internal customer relationship management (CRM) platforms. Furthermore, in most cases, even when an developer provided the categories of collected or shared personal information or the categories of third parties, sources, or purposes, these disclosures came directly from their privacy policies. We mark these cases as valid disclosures, as we are unable to verify whether those categories in fact apply to our case or not from the developer's response alone.

Response Format. The 69 companies that replied with the personal information collected about us communicated this information to us in a number of ways, including 23 (33%) companies that included the data directly in the email reply or as an email attachment, 19 (28%) that provided the data as an attachment on the VCR platform, and 12 (17%) that made it available from account or in-app privacy settings. The remaining 15 companies used a variety of methods to transmit the data to us, including, as a file shared with us via a cloud storage provider, as a download link in the email reply, or via a message sent to us through a customer support portal.

Security of the Process. We looked at the security mechanisms (if any) used by the developers of the 69 apps to securely communicate our personal information to us, beyond our email provider's access controls. At least 43 companies used an expiration time on the download links or files that they shared with us, ranging anywhere from 24 hours to 90 days. However, in 4 of these cases, we verified that the files remained accessible and downloadable even after the stated expiration time. Additionally, 26 app developers relied on their standard account authentication for access control, 2 used Gmail's "confidential mode" and 3 relied on other access controls, such as those enforced by cloud storage providers. Additionally, 16 companies required email verification to access and download the file, while 9 secured the data file by setting a password to open it, which they communicated separately to us.

Data Format. We looked at the format and characteristics of the 62 data files that contained specific pieces of collected personal information. Developers relayed the files using a number of formats, including CSV (27 instances), JSON (18), PDF (12), Excel (11), and TXT (9). Only 6 companies presented the same data using two different formats, whereas the remaining 56 either used a single format or a combination of several comprising a single data record.

Comparison with App Analysis Results

We strive to not only to understand the process of submitting a VCR under the CCPA, but also the accuracy of the data provided back to us. We first focus on the 68 companies who replied with the specific pieces of personal information. In this case, the response to the VCR included specific values that were collected by the developers, therefore, we simply matched the values from the VCR with the data that we observed being transmitted over the network.

Only 9 apps that provided us the specific pieces of personal information fully disclosed the extent of their data collection practices. With respect to the enumerated list of categories of personal information defined by the CCPA, we observed the collection, but not the disclosure, of identifiers by 55 apps, geolocation data by 21 apps, sensory data by 18 apps, customer record information by 16 apps and, to a lesser extent, professional information in 4 cases, characteristics of protected classifications (e.g., gender or age) in 3 cases, and education information in one case.

In terms of the specific pieces of personal information, we observed the collection, but not the disclosure, of device-specific identifiers, such as the Android Advertising ID (AAID), by 51 apps, app-specific identifiers, such as the Android ID, by 28 apps, coarse GPS coordinates (i.e., with a granularity up to a certain neighborhood) by 4, ZIP code by 8, the name of the city by 12 apps, precise GPS coordinates (i.e., that point to a specific building) by 12, parts of postal address by 10, user's phone number by 5, information about a user's contacts by 5 apps, and so on.

We examined the network transmission logs for the 8 apps developed by companies that told us that they did not hold any data on us; only one appeared to not actually collect any data. The remaining 7 collected data across a range of CCPA-defined categories of personal information, in particular, identifiers (7), geolocation (3), and sensory data (3). More specifically, all 7 apps collected the AAID, 5 collected our IP address, and one collected a device-identifying ID generated by the Branch.io SDK. Furthermore, one of the apps collected, but did not disclose the collection of precise GPS coordinates, and 3 apps collected coarse geolocation data that pinpointed the specific city, neighborhood, or ZIP code, where the device was physically located. Finally, 3 apps collected readings generated by the device's accelerometer, gyroscope, or magnetometer sensors.

Table 4.4 summarizes the undisclosed data collection that we observed across the 80 apps, for which we received a response, including information about the usage of TLS encryption,

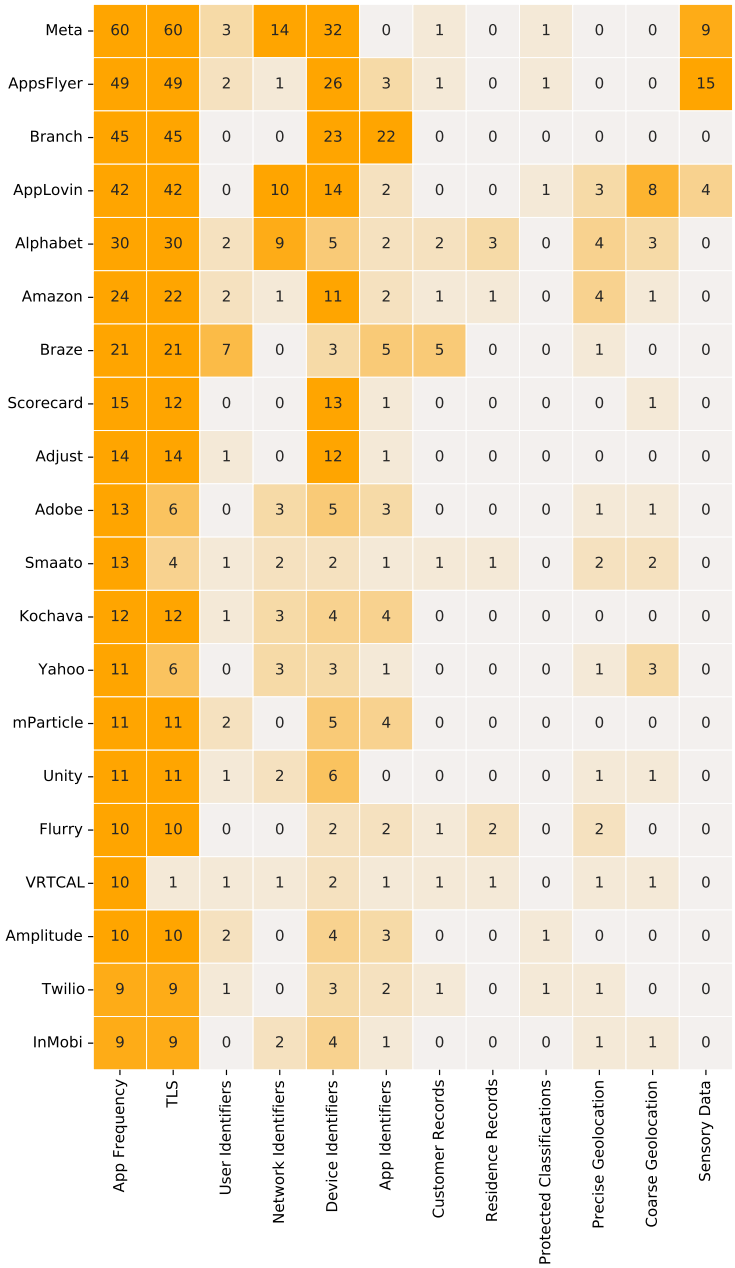

Figure 4.1: Top 20 third-party data recipients.

Category	**Subcategory**	**PII Name**	**#Apps**	**TLS #**	**1st Party #**	**3rd Party #**
Identifiers	User	Username	3	3 (100%)	0 (0%)	3 (100%)
	Network	IP Address	23	21 (91.3%)	9 (39.1%)	20 (87%)
		Router MAC	8	8 (100%)	2 (25%)	6 (75%)
		Router SSID	8	7 (87.5%)	3 (37.5%)	6 (75%)
	Device	AAID	49	44 (89.8%)	32 (65.3%)	43 (87.8%)
		Hardware ID	1	1 (100%)	0 (0%)	1 (100%)
		IMEI	3	3 (100%)	2 (66.7%)	1 (33.3%)
		IMSI	2	2 (100%)	1 (50%)	1 (50%)
		SIM ID	2	2 (100%)	1 (50%)	1 (50%)
		Wi-Fi MAC	1	1 (100%)	1 (100%)	0 (0%)
		Fingerprint ID	25	23 (92%)	2 (8%)	25 (100%)
	App	Identity ID	16	15 (93.8%)	1 (6.2%)	16 (100%)
		App Fingerprint ID	10	7 (70%)	8 (80%)	8 (80%)
		Android ID	20	17 (85%)	10 (50%)	18 (90%)
Customer Records	Customer	Phone Number	5	4 (80%)	5 (100%)	1 (20%)
	Contacts	Name	2	2 (100%)	2 (100%)	0 (0%)
		Phone Number	5	5 (100%)	5 (100%)	0 (0%)
	Residence	Street	3	3 (100%)	1 (33.3%)	2 (66.7%)
		City	5	4 (80%)	4 (80%)	3 (60%)
		County	2	2 (100%)	2 (100%)	1 (50%)
		ZIP Code	6	5 (83.3%)	5 (83.3%)	3 (50%)
Protected Classifications		Gender	1	1 (100%)	1 (100%)	1 (100%)
		Date of Birth	5	5 (100%)	4 (80%)	1 (20%)
Geolocation	Precise	GPS Coordinates	13	10 (76.9%)	10 (76.9%)	10 (76.9%)
	Coarse	GPS Coordinates	5	5 (100%)	3 (60%)	2 (40%)
		City	15	14 (93.3%)	8 (53.3%)	11 (73.3%)
		County	3	3 (100%)	3 (100%)	1 (33.3%)
		ZIP Code	9	7 (77.8%)	5 (55.6%)	7 (77.8%)
Professional		Job	2	1 (50%)	2 (100%)	1 (50%)
		Company	3	3 (100%)	3 (100%)	0 (0%)
Education		University	1	1 (100%)	1 (100%)	0 (0%)
Sensory Data		Sensor Readings	22	22 (100%)	1 (4.5%)	22 (100%)

'# Apps' denotes the total number of apps that did not disclose the specific PII out of a total of 80 apps that provided valid responses to VCRs. Percentages denote the proportion out of the total number of apps that did not disclose the specific PII.

Table 4.4: Number of apps that collected different categories of personal information without disclosure.

as well as the number of apps that do not disclose the categories of personal information shared with the first-party and third-party domains. We note that our results provide a lower bound on the number of pieces of collected-but-undisclosed personal information, as additional personal information collected by the apps might not have been detected during our analysis of the apps' network traffic. We additionally present the top 20 third-party recipients of personal information, as well as the number of apps that shared different categories of personal information with these entities in Figure 4.1.

Privacy Policies

Finally, we analyzed the disclosures made in the privacy policies of tested apps. For each of the 109 privacy policies containing CCPA-specific information, multiple researchers from our team independently indicated which categories of personal information were collected or disclosed by each developer and to which category of recipients. Table 4.5 summarizes the number of policies disclosing the collection and sharing of categories of personal information, the categories of recipients, and the inter-rater reliability scores.

All 109 policies disclosed the collection of identifiers and only two did not mention the collection of "Internet activity information," which includes data about app interactions. Additionally, 97 (89%) and 95 (87%) policies disclosed the collection of geolocation data and customer records information, respectively. The broad nature of these categories entails that most developers collect and, frequently, share this information, particularly in the context of mobile apps where technical identifiers, data from sensors, and usage information can be used both to provide the required app functionality and to track users. By the same token, users do not gain much by being informed about the collection of these categories.

We also identified the categories of personal information that the developers disclosed or sold,[9] as well as the categories of recipients of users' personal information. Although the CCPA requires companies to enumerate the recipients for each category of personal information, in practice we found that only a small number of policies did so. Therefore, we focused on locating the categories of recipients in the text of policies irrespective of which personal information they received.

Unsurprisingly, we found that the most frequently collected categories of personal information are also the most frequently shared. In particular, 103 (94%) policies disclosed the sharing of identifiers, 98 (90%) disclosed the sharing of internet activity information, and 84 (77%) disclosed the sharing of geolocation data. With respect to recipients, almost every privacy policy (106 or 97%) stated that the company might share users' personal information with law enforcement, if legally compelled. We also observed analytics providers (93% of policies), advertising networks (89%), and marketing partners (79%) being disclosed as the stated recipients of personal information from the apps' users.

[9]Cal. Civ. Code §1798.140(t)(1) broadly defines 'selling' as disclosing "a consumer's personal information by the business to another business or a third party for monetary or other valuable consideration," i.e., even when no monetary exchange is involved.

Prompt	Category	Yes #	No #	α
Does the privacy policy state that the app developer collects....	Identifiers	109	0	—
	Customer Records	95	14	0.517
	Protected Classifications	63	46	0.663
	Commercial Information	78	31	0.596
	Biometric Information	12	97	0.714
	Network Activity	107	2	0.176
	Geolocation Data	97	12	0.616
	Sensory Data	63	46	0.373
	Professional Information	46	63	0.726
	Education Information	15	94	0.616
	Inferences	62	47	0.542
Does the privacy policy state that the app developer discloses or shares....	Identifiers	103	6	< 0
	Customer Records	81	28	0.183
	Protected Classifications	49	60	0.411
	Commercial Information	65	44	0.445
	Biometric Information	8	101	0.579
	Network Activity	98	11	0.099
	Geolocation Data	84	25	0.287
	Sensory Data	53	56	0.275
	Professional Information	29	80	0.625
	Education Information	15	94	0.605
	Inferences	58	51	0.434
Does the privacy policy state that the app developer shares personal information with...	Affiliates	98	11	0.449
	Advertising Networks	97	12	0.356
	Marketing	86	23	0.517
	Analytics	101	8	0.275
	Security and Fraud	66	43	0.293
	Payment Processors	78	31	0.573
	Customer Support	55	54	0.596
	Storage and Infrastructure	59	50	0.637
	Search Engines	10	99	0.347
	Social Media	49	60	0.599
	Order Fulfillment	25	84	0.559
	Law Enforcement	106	3	0.234
	Unspecified Partners	78	31	0.042

Column 'α' refers to Krippendorff's alpha, a measure of inter-rater reliability.

Table 4.5: We observed information practices relevant to these categories of personal information in privacy policies.

For many categories, we did not attain a significant level of inter-rater reliability (Krippendorff's α in Table 4.5). We attribute this result to the broad nature of some categories. For instance, there is a significant overlap between the 'identifiers' and 'customer records' categories. Recipients of personal information also commonly fall into similar categories, e.g., many companies that provide advertising also offer analytics and marketing solutions. Finally, although some companies used the CCPA-defined categories of personal information to describe their data collection and sharing practices, others relied on their own categorizations, and the CCPA does not define the categories of third-party recipients, further decreasing the consistency between policies written by different developers.

We observed the highest inter-rater agreement regarding the collection of professional or employment-related data (Krippendorff's $\alpha = 0.726$), biometric data (0.714), and protected classifications (0.663). In general, a Krippendorff's alpha of .667 or higher is considered acceptable for drawing tentative conclusions [118].

Categories Comparison. Finally, we compared the categories of personal information that we observed being collected and the categories of recipients with the categories disclosed by the developer in the VCR response and with the categories that we obtained from analyzing the privacy policies. We present the results of this comparison for the 80 apps that completed the VCR in Table 4.6. Compared to the VCR responses, 25 (31%) privacy policies failed to fully inform us about all of the categories of collected personal information, while only 17 (21%) did not fully disclose the sharing of information to third parties.

4.5 Discussion

Our results present several important implications for developers and policy makers with respect to the process of submitting verifiable consumer requests and ensuring accurate responses. We highlight the following areas for improvement: determining CCPA applicability, the security of consumers' personal information, and the usability, completeness, and accuracy of developers' responses.

Determining CCPA Applicability

Only 71% of selected apps included CCPA-specific disclosures in their privacy policies. As a compromise between evaluating the compliance of popular apps without burdening smaller developers that do not have to comply, we decided only to submit VCRs to those who provided CCPA-specific information in their privacy documents. However, this naturally limited the scope of our analysis and also prompted us to consider how ordinary consumers could determine which companies are covered by CCPA requirements.

We imagine that the only organizations that consumers could realistically determine to conform to the CCPA's definition of a "business" (see Section 2.1) are public companies

Categories	Apps	Policies		VCRs	
		Disclosed	*Undisclosed*	*Disclosed*	*Undisclosed*
Identifiers — Collection	75	74	1	22	53
Identifiers — Sharing	60	55	5	11	49
Customer Records — Collection	59	56	3	12	47
Customer Records — Sharing	16	13	3	2	14
Protected Classifications — Collection	16	9	7	5	11
Protected Classifications — Sharing	4	2	2	1	3
Geolocation Data — Collection	38	36	2	6	32
Geolocation Data — Sharing	23	20	3	3	20
Sensory Data — Collection	22	15	7	0	22
Sensory Data — Sharing	22	10	12	0	22
Professional Information — Collection	12	5	7	3	9
Professional Information — Sharing	1	0	1	0	1
Education Information — Collection	8	2	6	1	7
Education Information — Sharing	0	0	0	0	0
Affiliates or Subsidiaries	3	3	0	0	3
Advertising Networks	23	22	1	5	18
Marketing	27	17	10	3	24
Analytics	49	46	3	7	42
Security and Fraud	3	3	0	0	3
Payment Processors	2	2	0	0	2
Customer Support	1	0	1	0	1
Storage and Infrastructure	26	15	11	2	24
Search Engines	5	0	5	0	5
Social Media	35	15	20	1	34

'Apps' denotes the number of apps observed collecting or sharing a specific category of PII, or the number of apps that transmitted some PII to a specific third-party recipient, while 'Disclosed' indicates how many of these disclosed that collection or sharing in a privacy policy or a VCR.

Table 4.6: Number of apps we observed collecting or sharing a specific category of personal information and the number of privacy policies that disclosed these information practices.

that disclose revenues in earnings reports. However, this severely limits the ability of consumers to determine whether a company has to comply with the CCPA; even if everyone could easily read earnings reports, fewer than 0.01% of companies in the U.S. are publicly traded [63]. Companies with a large online presence can surpass the data collection threshold if, for instance, they use cookies, other tracking technologies, or even simply record technical information from users' devices, such as IP addresses, but there is no way for consumers to know when the threshold is met. This could be addressed by requiring all companies doing business in California to state in their privacy policies whether they are subject to the CCPA.

Authentication and Security

Our analysis also demonstrated that many app developers did not use any identity verification mechanism beyond a proof of access to an email account; other companies required copies of government-issued identity documents and signed affidavits. Given different domains and company sizes, it is unlikely that a one-size-fit-all authentication approach will work for all organizations. However, we highlight several issues that we encountered and propose solutions.

For apps that maintain user accounts, we suggest relying on existing authentication mechanisms to submit requests and access the provided data. At the very least, these companies should require a password to perform these actions. Ideally, these companies would also require a second authentication factor, such as a mobile push notification or a one-time password (OTP). App developers should also notify users about VCR submissions using established communication channels to help detect fraudulent requests.

Authentication is more difficult for developers that do not require the creation of user accounts to access their apps. These companies should request at least three (and possibly more) non-trivial pieces of user-specific information to match against the data already held. In the case of mobile apps, the developer could require the user to send the VCR via the app, such that the request also contains device-specific information alongside the requested user-specific information. However, developers should also provide an option to submit VCRs via other means, as a user might have already uninstalled the app or changed their device. If the company does not hold sufficient information to verify the consumer to a reasonable degree, then they should rightfully reject the request to avoid leaking consumers' personal information to unauthorized parties. Companies should also **not** request copies of government-issued IDs for authentication, as most organizations would not (and, ideally, should not) have access to unique ID numbers to match against; information in photos, such as name or birthdate, can be easily digitally altered.

Finally, once the developer successfully confirms the identity of the consumer, they should take necessary precautions to secure access to and transmission of consumer's personal information. In addition to existing authentication mechanisms and, ideally, two-factor authentication, developers should employ TLS, use download links with a time expiration, and secure

files using a password set by the consumer beforehand. Although none of these measures can fully prevent the leakage of personal information, they can definitely increase the cost for attackers attempting to fraudulently gain access to consumers' sensitive information.

Usability, Completeness, and Accuracy

We also discovered that VCR responses from app developers noticeably varied in their format and contents. For instance, although 97% of companies that completed our requests provided specific pieces of personal information, that proportion dropped to 35% for categories of third parties. Furthermore, only 7 companies provided a choice to receive the data either in a human-readable (e.g., TXT) or a machine-readable format (e.g., JSON).

We believe that regulators should issue more guidance to businesses when it comes to the logistics of providing personal information back to consumers. Besides questions of authentication and security, regulators should provide examples of categorizations that developers could use in responding to VCRs. For instance, although the text of the CCPA mentions covered categories of personal information, similar categories for third parties or sources of collection are absent. Many businesses use CCPA-defined categories of personal information in their policies and VCR responses and, thus, similar taxonomies would be beneficial in other contexts. We believe that to achieve greater transparency, the CCPA should also require companies to disclose names of third parties with whom they share personal information, as opposed to only requiring the categories to be disclosed.

With respect to the accuracy of responses containing specific pieces of personal information, we discovered that developers would often collect but not disclose identifiers, geolocation data, and sensory data. As is already the case in newer versions of Android, developers should not be allowed to collect persistent non-resettable identifiers from consumers' phones, such as hardware identifiers. Instead, developers and third-party libraries should only gain access to dedicated, resettable identifiers, specifically, the Android Advertising ID (AAID). Regulators should also remind developers that device identifiers, even resettable ones, constitute personal information under the CCPA and, therefore, have to be disclosed upon receipt of a verifiable consumer request. Developers should also be reminded that the collection of such identifiers increases their chance of becoming subject to the CCPA once they reach the predefined data collection threshold. Providing more examples to developers, especially in the context of mobile apps, could help clarify what information and at which level of granularity constitutes personal information under the CCPA.

Finally, the CCPA's "right to know" encompasses two distinct privacy rights: the right of access and the right to data portability. Although both rights can provide access to personal information held by a business, they serve different purposes. Whereas data provided under the right to data portability should be easily imported or transmitted to another service, data provided under the right of access should be comprehensible to the consumer to whom the data pertains. As these two privacy rights are not differentiated under the CCPA the same way they are, for instance, under the GDPR, businesses provide responses mainly in the

machine-readable formats that are easier to export, such as JSON. However, such formats are unlikely to be easily usable by ordinary consumers. We therefore argue that the CCPA could be enhanced by differentiating between the two rights and by providing guidelines to developers about the best practices and formats to use when responding to requests under each of these rights.

4.6 Limitations

We investigated the extent to which Android app developers comply with the provisions of the CCPA that require them to disclose their data sharing practices in privacy policies and in response to consumers' access requests. As our objective was to select apps that we reasonably inferred to fall under the CCPA definition of a "business," it is important to note that the resulting sample of apps is not meant to be representative. Our results, therefore, do not generalize to the entire population of Android apps and do not necessarily provide insights about the data collection and sharing behaviors of other apps.

As we previously explained in Section 4.3, we tested the apps and interacted with developers using pseudonyms. We acknowledge that some companies may not have automated systems to process CCPA-related requests, and therefore processing our VCRs may have imposed costs on the employees responding to requests. However, as in related studies [18, 99, 144, 120, 209, 226], we believe that our approach was necessary to investigate the quality of the VCR responses under realistic conditions and to mitigate research participation effects [135]. Furthermore, we believe that that business interests in this regard are outweighed by the public interest in understanding the effectiveness of CCPA rights and raising awareness around existing issues.

Finally, the developments in privacy regulation will necessitate further work in understanding how changes in specific scopes and provisions translate into differences in compliance of different businesses. In particular, most of the provisions of the California Privacy Rights Act (CPRA) revising the CCPA will become operative on January 1, 2023, with enforcement commencing on July 1, 2023. We believe that future work should continue examining the application of and compliance with the new privacy regimes to guide the development of further consumer data protection laws.

Chapter 5

How do App Developers Implement Privacy Protections?

Like most modern software, secure messaging apps rely on third-party components to implement important app functionality. Although this practice reduces engineering costs, it also introduces the risk of inadvertent privacy breaches due to misconfiguration errors or incomplete documentation. Our research investigated secure messaging apps' usage of Google's Firebase Cloud Messaging (FCM) service to send push notifications to Android devices. We analyzed 21 popular secure messaging apps from the Google Play Store to determine what personal information these apps leak in the payload of push notifications sent via FCM. Of these apps, 11 leaked metadata, including user identifiers (10 apps), sender or recipient names (7 apps), and phone numbers (2 apps), while 4 apps leaked the actual message content. Furthermore, none of the data we observed being leaked to FCM was specifically disclosed in those apps' privacy disclosures. We also found several apps employing strategies to mitigate this privacy leakage to FCM, with varying levels of success. Of the strategies we identified, none appeared to be common, shared, or well-supported. We argue that this is fundamentally an economics problem: incentives need to be correctly aligned to motivate platforms and SDK providers to make their systems secure and private by default.

In the previous chapter, we presented our first empirical study on the failures of popular Android app developers to successfully implement the privacy rights of access and transparency. This chapter[1] applies the measurement framework we established in Chapter 3 to measure the failure of app developers to operationalize the privacy requirements defined by the underlying data protection principles of *data minimization* within the context of secure messaging apps.

[1]This technical chapter is based on work previously published in a peer-reviewed journal [167].

5.1 Introduction

Modern economies rely on the specialization of labor [177]. Software engineering is no different: modern software relies on myriad third-party components to fulfill tasks so that developers do not need to waste time rebuilding specific functions from scratch [74]. This type of "code reuse" is a recommended practice and transcends many branches of engineering (e.g., car manufacturers do not manufacture every component that goes into their cars, instead relying on components from third-party suppliers). Software development kits (SDKs) facilitate code reuse during software development and offer many benefits for developers. They provide well-trodden paths: documented workflows for developers to follow so that these developers can consistently provide common functionality. Ultimately, SDKs reduce engineering costs when used responsibly.

Yet, recent research has demonstrated that many software privacy issues (i.e., the inappropriate disclosure of sensitive user information) are due to developers' misuse of third-party services [163, 9]. That is, privacy breaches often occur due to developers not correctly configuring SDKs, not reading SDK documentation, or SDKs behaving in undocumented ways, often unbeknownst to developers. This is especially concerning when the third-party SDK may transmit highly sensitive user data to third parties and the SDK is ubiquitous across many software supply chains.

Heightened public concerns around the monitoring of online communications have significantly influenced consumer behavior in the past decade. A 2014 PEW survey found that 70% of Americans are concerned about government surveillance and 80% about surveillance by corporations [131]. In response to these concerns, more and more consumers have begun using secure messaging apps to protect their communications based on the promises of privacy made by these apps. Hundreds of millions of users now use apps like Signal or Telegram, believing these apps to protect their privacy. These applications are entrusted with a vast array of confidential user data, from personal conversations to potentially-sensitive multimedia content, thereby placing a significant emphasis on their ability to make good on their promises of privacy and security.

The misuse of third-party SDKs within secure messaging apps may pose a heightened risk to users because those SDKs may leak sensitive information to third parties. In particular, app developers use third-party SDKs to implement *push notifications*, which display important information to the user, including messages from other app users (Figure 5.1). Because push notification SDKs are generally provided by third parties (as opposed to app developers), incorrect usage may leak sensitive information to those third parties. For example, an app that provides "end-to-end" encrypted messaging may not actually provide end-to-end encryption if message payloads are not encrypted before being sent to third-party push notification APIs. To make matters worse, misuse of these SDKs may also contribute to the misrepresentation of security and privacy assurances to consumers as articulated in various disclosures, including privacy policies, terms of service, and marketing materials.

The combined risk of sensitive information leakage and misrepresentation of privacy

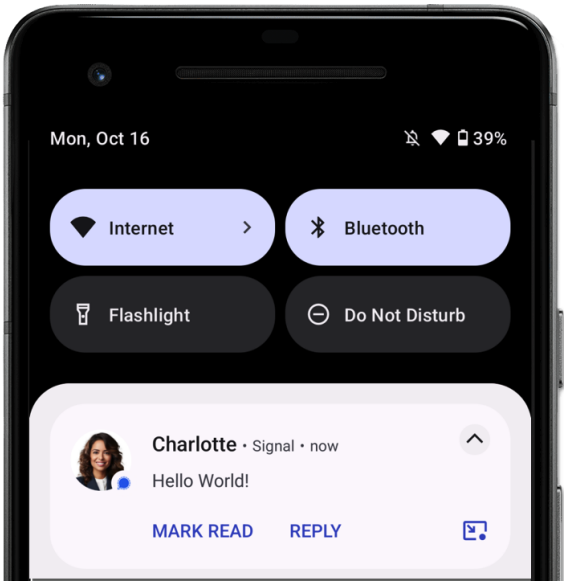

Figure 5.1: An illustration of an Android push notification.

promises creates serious ramifications for users of secure messaging platforms. Oppressive regimes or other adversaries may use court orders to compel companies involved in the delivery infrastructure of push notifications to reveal the contents of communications sent and received by human-rights workers, political dissidents, journalists, etc. Worse, when this does happen, both the developers of the apps and the users who are endangered are unlikely to be aware that their communications are being intercepted. This threat model is not just theoretical. Crucially, since we performed our analysis, U.S. Senator Ron Wyden published an open letter that confirms that government agencies do, in fact, collect user information by demanding push notification records from Google and other push notification providers through the use of legal processes [228]. Our work is highly prescient, as it provides new insights into an emergent threat model.

To study the extent to which the delivery infrastructure may access sensitive user information, we examined the use of Google's *Firebase Cloud Messaging* (FCM) to deliver push notifications to *secure messaging apps* on Android devices. Google provides FCM as a free service, and therefore, it is one of the most commonly used third-party SDKs to deliver Android push notifications. Moreover, the majority of other push services, including OneSignal [146], Pusher [158], and AirShip [6] internally rely on Google's FCM to deliver notifications to Android devices, making the usage of FCM practically unavoidable for developers who wish to provide push notification support in their Android apps. (On Apple's iOS, third-party push notification APIs are similarly built on top of Apple's push notification service [147].) We focus on secure messaging apps because these apps (1) market their abilities to keep message data "private" or "secure" and (2) make heavy use of push

notifications to notify users of incoming messages and their contents (and therefore, when not implemented correctly, may run the risk of leaking message contents and metadata to the push notification service).

Prior work has investigated the potential security risks that push notifications may pose, including by push notification-based malware [101, 125] and botnets [122, 101]. To our knowledge, no work has focused on the privacy risks of push notification services used by secure messaging apps. Therefore, we performed a study to examine whether the push notification records potentially stored without end-to-end encryption by the delivery infrastructure may misrepresent or compromise the privacy protections of secure messaging and expose users to legal risks. Thus, we posed the following research questions:

- **RQ1:** What personal data do secure messaging apps for Android send via Google's Firebase Cloud Message (FCM)?
- **RQ2:** What mitigation strategies do app developers use to protect personal information from being disclosed to Google's FCM?
- **RQ3:** Do the observed data-sharing behaviors align with the privacy assurances apps make in their public disclosures?

To answer these questions, we performed static and dynamic analysis on a corpus of 21 secure messaging apps. We used dynamic analysis to understand what data these apps sent over the network. When we found that apps displayed data in push notifications, but did not obviously send that data over the network, we used static analysis to understand what mitigation strategies they used to achieve this effect. In contrast, when segments of data displayed in the app *were* verbatim in push notifications, we further examined these messages to assess whether sensitive data was available in plaintext to the delivery infrastructure. Finally, we analyzed apps' privacy policies and other disclosures to identify the privacy claims that apps made to users. By comparing observed behavior from our app analysis to disclosed behavior, we identify undisclosed sharing and potentially-misleading data practices: data that apps imply that they will not disclose, but—intentionally or not—do disclose to the delivery infrastructure through the use of push notifications.

We found that more than half of the apps in our corpus leak *some* personal information to Google via FCM. Furthermore, none of the data we observed being leaked to FCM was specifically disclosed in those apps' privacy disclosures. We also found several apps employing strategies to mitigate this privacy leakage to FCM, with varying levels of success. Of those identified strategies, none appeared to be common, shared, or well-supported. While app developers are ultimately responsible for the behavior of their apps, they are often ill-equipped to evaluate their apps' privacy and security properties in practice. Given that the problems that we observe are pervasive across app developers and stem from the use of third-party components that can be easily used insecurely, we conclude that SDK providers are best positioned to fix these types of issues through both better guidance and privacy-preserving designs and defaults.

In this paper, we contribute the following:

- We demonstrate the widespread sharing of personal information, perhaps inadvertently, with Google through developers' use of push notifications.
- We highlight systemic mismatches between privacy disclosures and observed behaviors in delivering push notifications via FCM.
- We discuss developers' negligence in deploying software that they do not understand and the responsibility that SDK and platform providers share in creating infrastructures that are private/secure by default.

5.2 Background

We provide an overview of push notification services (PNS), specifically Google's Firebase Cloud Messaging (FCM). We describe the threat model we consider in this paper and our overall motivation.

Mobile Push Notifications

A push notification is a short message that appears as a pop-up on the desktop browser, mobile lock screen, or in a mobile device's notification center (Figure 5.1). Push notifications are typically opt-in[2] alerts that display text and rich media, like images or buttons, which enable a user to take a specific action in a timely fashion, even when the app in question is in the background. Applications often use push notifications as a marketing or communication channel, but they can also be used as a security mechanism (e.g., as part of a multi-factor authentication ceremony).

There is a difference between push messages and notifications. "Push" is the technology for sending messages from the server-side component of the app (the "app server") to its client side (the "client app"), even when the user is not actively using the app. Notifications refer to the process of displaying timely information to the user by the app's user interface (UI) [22]. In the context of mobile apps, the application server can send a push message without displaying a notification (i.e., a silent push); an app can also display a notification based on an in-app event without receiving any push messages. For simplicity's sake, we use the term "push notifications" in this paper regardless of whether an actual notification is displayed to the end user (i.e., we refer to messages flowing through a cloud messaging server to a user's device, whereupon the device's operating system routes the messages to the appropriate app).

Although app developers could, in theory, implement their own push notification service, this is usually impractical as it requires the app to continually run as a background service, thereby reducing battery life. Instead, most mobile app developers rely on *operating system*

[2]Android and iOS require user permission before an app can display notifications.

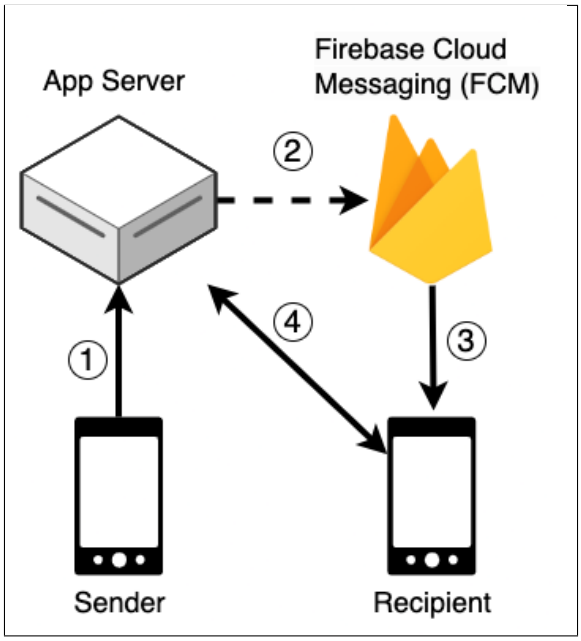

Figure 5.2: Flow chart of FCM's push notification infrastructure for messaging apps.

push notification services (OSPNSs), including Firebase Cloud Messaging (FCM) for Android or Apple Push Notification Service (APNS) for iOS devices [13]. FCM and other PNSs facilitate push notifications via an SDK the developer adds to their application. When a user launches the app for the first time, the SDK registers the device with the PNS by generating a *push token* (also known as a *registration token*), which serves as a pseudonymous identifier that tells the push service where to forward the messages. The SDK returns the push token to the client app, which should then be sent and stored in a database on the app server. When the app wants to send a push notification, it looks up the appropriate push token and sends it alongside the message to the PNS, which then forwards the message to the correct device [221]. The push token is tied to the app instance, and therefore, the developer should periodically refresh it, e.g., if the user deletes and reinstalls the app.

In sum, there are three main actors involved in the process of sending push notifications using FCM (see also Figure 5.2):

App Server sends event-specific messages to FCM (2). For instance, in the context of a messaging app, a sender device may send a message to the app server (1), which then sends a push notification request to FCM (2).

Firebase Cloud Messaging (FCM) is a cloud-based OSPNS that forwards push messages to the appropriate user device using the stored registration token(3), even if the

client app is offline or in the background. It also exposes an API to the developer to enable push messaging in their applications.

Android Device runs the OS and the client app. Android uses a system component that is part of Google Play Services to receive push messages sent by FCM, which it then passes to the appropriate app. Optionally, the client app can also query additional information from the app server (4) in response to a received push notification.

The SDKs distributed by FCM and other PNSs not only streamline app development by reducing the amount of code that needs to be written, but in many cases, their use is necessary for performance and efficiency reasons [193]. Developers would also need to request the Android permission for unrestricted battery usage, something a user might not necessarily grant. As such, mobile platform owners only provide official support for their managed OSPNSs: Google for FCM and Apple for ASPNS.[3]

FCM Alternatives

Given the utility of push notifications, companies have started offering push notification services that compete with Google's FCM. These third-party PNS providers, such as Airship, Pushwoosh, and Taplytics, may offer advantages over FCM, including more features or usable APIs. While it may seem that developers using third-party PNSs can potentially avoid the security and privacy pitfalls of FCM, Lou et al. demonstrated that third-party push providers rely on FCM to deliver messages to Android devices with Google Play Services [129]. The authors identified the dual-platform structure of push notifications. The first service ("host notification platform") abstracts push messaging by providing an API that interfaces with the second service ("transit notification platform"), which provides a stable system-level communication channel to deliver push notifications to user devices. While both FCM and third-party PNSs offer developer-facing APIs for managing push notifications (i.e., the host notification platform), only FCM fulfills the role of the transit notification platform and delivers messages internally to Android devices with Google Play Services.

Furthermore, we found statements by several popular third-party PNSs, such as OneSignal [146], Pusher [158], and AirShip [6] that mention their dependence on FCM for sending push notifications to Android devices. For instance, OneSignal states in a blog post that "Google mandates that Android apps distributed through Google Play leverage a single, shared connection provided by FCM" and "OneSignal itself uses the FCM API internally to send messages to Android devices" [146]. Therefore, these third-party PNSs expose users to risks associated with FCM push notifications while potentially introducing their own problematic data collection practices. For instance, Reuters has previously reported that Pushwoosh—a third-party PNS—misrepresented itself as based in the U.S. despite actually

[3]We studied Android because the operating system is open source, allowing us to more easily build instrumentation to monitor app execution.

being headquartered in Russia [152]. Although Pushwoosh denied the claims [115], the revelation still led the U.S. Army and Centers for Disease Control and Prevention (CDC) to stop using apps containing the Pushwoosh SDK.

Android devices without preinstalled Google Play Services either do not properly support push notifications or use an alternative platform. Most notably, Android devices sold in China do not include Google Play Services, but use another preinstalled service provided by the phone manufacturer, such as Huawei Mobile Services (HMS), to handle push notifications. There are other Android variants outside of China that do not come with Google Play Services preinstalled, such as FireOS, which runs on Amazon devices and uses Amazon Device Messaging (ADM) instead of FCM. These variants constitute a small share of the global Android market [79] and are outside the scope of our analysis.

Other alternatives, such as UnifiedPush [205] or Samsung Push Service [65], rely on apps to receive push notifications in place of Google Play Services. However, we argue that such solutions do not represent equivalent alternatives, as they require users to install an additional app and developers may still use FCM as the push service, unbeknownst to app users. Thus, we specifically focus on data shared with Google's FCM, regardless of the specific third-party service running on top of it. (That is, our instrumentation is agnostic as to whether it captured messages sent natively using FCM or another third-party API built upon it.)

Threat Model

FCM acts as an intermediary between the server-side and client-side applications and uses push tokens to identify the device where push notifications should be forwarded. While efficient, this architecture poses three significant privacy risks to users [227, 55]:

Disclosure. The contents of a push notification and its metadata may be disclosed to unauthorized entities.

Linking. Push tokens may be linked or attributed to specific users or behaviors.

Identification. Individuals may become identified based on the information linked to their device's push tokens.

The primary threat model that we consider is the use of legal processes to request FCM push tokens linked to a targeted device and stored by the app developer. In the context of secure messaging apps, knowing the pseudonym (i.e., username) of the targeted user may suffice. Even if the app developer does not collect other identifying personal information, they must still store registration tokens to route the push notifications to the user's device through FCM servers. After obtaining the push tokens from the app publisher, law enforcement can request that Google provide all information linked to the given push token, which may

include the contents and metadata of the associated push notifications. Combining these pieces of personal information increases the risk of identification.

This threat model is not theoretical. In December 2023, U.S. Senator Ron Wyden published an open letter confirming that government agencies collect user information by demanding push notification records from Google and Apple through legal processes [228]. Since then, journalists found more than 130 search warrants and court orders going back to 2019 (e.g., [206, 207, 54]) in which investigators had demanded that tech companies, notably Wickr and TeleGuard—both advertised as end-to-end encrypted secure messaging apps—turn over push tokens associated with accounts of specific users. In the case of TeleGuard, an FBI agent then asked Google to hand over all information connected to the push token, which Google responded to with account names and IP addresses associated with those accounts [98]. Furthermore, Apple disclosed in its transparency report for the second half of 2022 that it received 70 requests worldwide seeking identifying information about Apple Accounts (formerly known as Apple IDs) associated with 794 push tokens and provided data in response to 54 (77%) requests. Google does not specifically break out government requests for push notification records and, instead, reports these requests in aggregate with other account data requests [14].

We hypothesize that many Android app developers transmit sensitive information via established third-party push notification channels and do not realize that they are not properly securing it. In a departure from "privacy-by-design" principles [38], the official Google Android Developers Blog recommends [172] that developers using Google's service "send as much data as possible in the [push notification] payload" and fetch the remainder of the data from the app server if needed. In the next paragraph of the blog, developers are advised that they "can also encrypt FCM messages end-to-end using libraries like Capillary," thereby indicating that FCM does not encrypt payload data by default (i.e., developers need to rely on additional libraries). There is no other mention of end-to-end encryption in the blog. Thus, questions remain as to whether developers follow this optional guidance.

Google's FCM developer documentation [84] states that "depending on your needs, you may decide to add end-to-end encryption to data messages" and "FCM does not provide an end-to-end solution." No further guidance is given on what information is appropriate to send. In contrast, Apple's documentation for sending notifications [13] instructs developers not to include "customer information or any sensitive data in a notification's payload" and, if they must, "encrypt it before adding it to the payload." Even if the majority of data sent using push notification channels is not personal, there are examples in which it might be, such as some user-generated content in instant messaging apps or sensitive information sent by a banking or a health-tracking app. In these cases, app vendors may be held liable for failing to safeguard or minimize the amount of personal information sent via push notification servers and for failing to disclose this practice in their privacy notices.

Given FCM's role as an intermediary, we posed the question: do apps leak user information through push notifications to the delivery infrastructure? We investigated this question by performing both mobile app analysis and analysis of privacy disclosures.

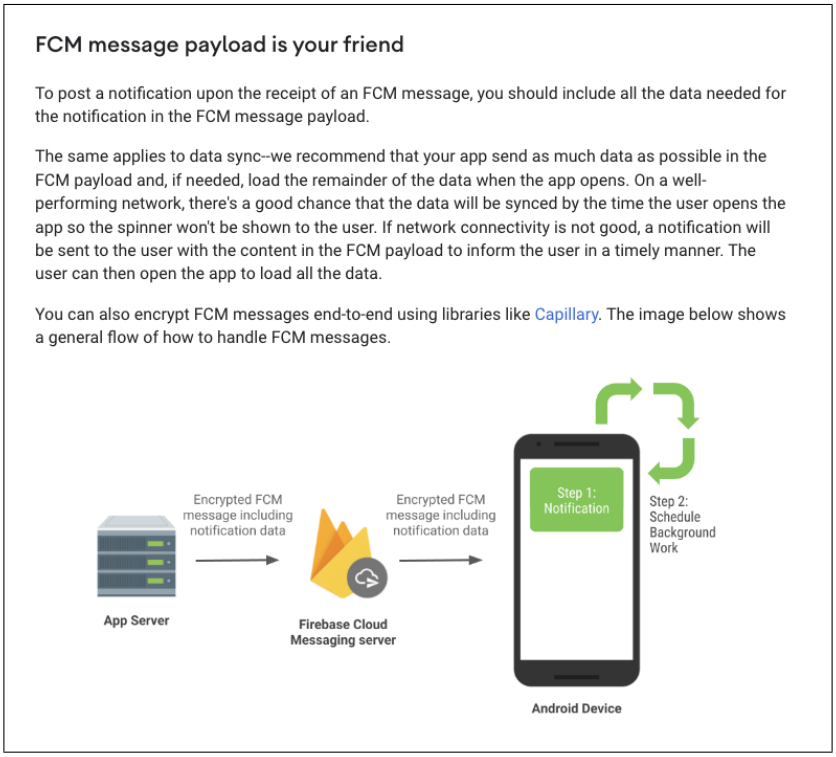

Figure 5.3: Google's guidance to send as much data as possible via FCM payloads.

5.3 Related work

In this section, we provide an overview of related work on the privacy and security risks of push notifications, mobile app analysis, and analysis of privacy-relevant disclosures.

Risks of Push Notifications

Prior research has demonstrated how attackers can exploit mobile push notifications to spam users with advertisements [127], launch phishing attacks [230], and even issue commands to botnets [5, 122, 101]. Other studies have revealed additional security issues with PNSs that can result in the loss of confidentiality (i.e., user messages get exposed to unauthorized parties) and integrity (i.e., users receive malicious messages from unauthorized parties) [43]. By assuming that the victim installs a malicious app, prior work has demonstrated how attackers can abuse platform-provided OSPNSs, including Google's FCM (formerly known as Google Cloud Messaging or GCM, and Cloud to Device Messaging or C2DM prior to that), to steal sensitive messages or even remotely control the victim's device [125]. Warren et al. described "security" as a key nonfunctional requirement for implementing push notification mechanisms and identified the push-to-sync strategy back in 2014 (which they called "poke-

and-pull") as a viable protection strategy for protecting user data from PNSs [217].

As described previously (§ 5.2), push notification architecture can be separated into the host platform that provides the push API and the transit platform that actually delivers the push notification internally. Several studies looked at the security issues of third-party PNS SDKs while excluding system-level transit platforms, such as FCM from Google. One study analyzed 30 different third-party PNS SDKs embedded in 35,173 Android apps and found that 17 SDKs contain vulnerabilities to the confidentiality and integrity of push messages, which an attacker can exploit by running a malicious app on the victim's device [43]. Similarly, Lou et al. performed a security and privacy analysis of the twelve most popular PNSs and compared their behavior in 31,049 apps against information practices disclosed in the privacy policies of those PNSs [129]. They found that out of twelve third-party PNSs, six PNSs collect in-app user behavior and nine collect location information, often without awareness or consent of app users. As the authors focused only on the host platforms, their analysis excluded FCM (and other transit platforms) on the basis of them being a "trustful service provider." We complement this work by focusing instead on the privacy risks of transit notification platforms, in particular, FCM from Google.

In recent years, researchers have analyzed PNSs from the perspective of privacy protection goals that complement the classic "CIA triad" (confidentiality, integrity, and availability), such as unlinkability, transparency, and intervenability [95]. One study, for instance, considered an adversary with the capability to silently sniff packets directed to or from the victim and actively trigger push notification messages to the target's personal device [128]. The authors demonstrated that under these assumptions, an adversary on the same network can deidentify the victim even if they use an online pseudonym. We complement these studies by focusing on FCM privacy risks in the context of secure messaging apps.

5.4 Methodology

Our primary research question concerns how secure messaging apps' usage of FCM impacts user privacy. To answer this question, we identified a set of apps from the Google Play Store and compared the claims made in their privacy disclosure documents with our static and dynamic analysis of those same apps.

The diagram in Figure 5.2 outlines the main actors and communications involved in push notification usage in secure messaging apps. The messaging app is installed on the phone/device of the sender and the recipient. First, the sender composes their message, and some content gets sent over the network to the app's server (1). Then, the server uses the FCM API to construct the push notification with the required payload. The FCM API sends the notification to Google's FCM server (2), which then forwards it to the recipient device (3) using a long-lived TCP connection initiated by Google Play Services. Finally, the data is parsed and packed into an intent that is then broadcast to the app, which displays the message in the form of a notification. Inadvertent data leakage to Google occurs when the

server places user information as plaintext in the push notification payload. Crucially, users and developers are likely unaware that Google may receive and, sometimes, retain[4] message contents and other metadata associated with the push notification.

As highlighted in § 5.3, numerous prior works evaluate the security and privacy of end-to-end (e2e) encryption and its implementation in secure messaging apps, including many of the ones in our corpus. However, our work is explicitly **not** investigating these claims of e2e encryption. Therefore, we are not interested in recording the traffic sent over a network connection. Rather, our interest is in determining whether implementing push notification functionality in a given app leaks personal message content to parties *other than the app developer*, specifically to Google via FCM. Therefore, we are primarily interested in what data the app's server sends to FCM via network connection. However, because we are out-of-band from this network connection, the best alternative is to record the inbound/outbound traffic on the recipient's device to infer which data may have been sent from the server to FCM. If the sender's plaintext message content is present in the push notification sent to the recipient's device from FCM, then it is clear that the app server did leak the user's message content to FCM. However, if the push notification sent to the recipient's device does not contain the sender's plaintext message, then it may be likely that the app server did **not** leak the user's message content to FCM.[5] For apps that fall into this category, we additionally want to understand the techniques they leverage to avoid leaking user message content and metadata to FCM.

App Selection

We selected messaging apps that made claims about the privacy of users' messages (herein, "secure messaging apps"). For example, Telegram's homepage promotes its app as "private" and states that "Telegram messages are heavily encrypted" [192]. Similarly, Signal's homepage encourages people to "speak freely" because the Signal app has a "focus on privacy" [174]. Signal publicly writes about what data their app collects and the fact that—in response to a legal subpoena requesting a range of user information—Signal is only able to provide "timestamps for when each account was created and the date that each account last connected to the Signal service" [173]. WhatsApp also explicitly markets the privacy benefits of their app and states, "[y]our privacy is our priority. With end-to-end encryption, you can be sure that your personal messages stay between you and who you send them to" [194, 195]. Because secure messaging apps make these claims about the privacy of users' messages, many users utilize these apps in sensitive contexts. For example, Telegram, Signal, and WhatsApp, three of the apps we analyzed, are frequently used by protesters worldwide [176, 211]. The apps in our data set, a subset of all secure messaging apps, are widely used and encompass over 2.8 billion users and 6.1 billion installs.

[4] E.g., FCM servers retain messages by default when the recipient device is offline.

[5] If the app server has access to the sender's plaintext message, then it is always possible that it is leaked to third-parties in ways that are not externally detectable, since traffic between the app server and these third parties is not observable.

Material Representations. The selection of messaging apps based on their privacy claims is not only a prudent approach for users prioritizing the confidentiality of their communications, but also a legally-grounded strategy, reflecting the enforceable nature of such assertions. When companies publicly assert their services' privacy and security features, these claims become material representations that can significantly influence consumer choices. Importantly, material misrepresentations are actionable under consumer protection laws. For instance, under the FTC Act[6] (and various state consumer protection laws), businesses in the U.S. are prohibited from materially misrepresenting their practices to consumers. The Federal Trade Commission (FTC) and state attorneys general actively monitor and pursue companies that fail to uphold their privacy promises (regardless of whether they are made in privacy policies [46] or marketing materials [45]). This enforcement protects consumers and reinforces the message that privacy and security assertions are material representations that have legal consequences and can affect consumer choices.

One such notable case is that of Zoom, in which the company faced a regulatory enforcement action for erroneously claiming to offer end-to-end encryption in its marketing materials, a feature it did not fully provide at the time [71]. This incident underscores the seriousness with which authorities treat misrepresentations in the digital privacy domain, highlighting the risks companies face when they do not accurately describe their data protection measures. Thus, evaluating messaging apps based on their stated privacy features is not only a measure of their utility in sensitive contexts, but also an assessment of their compliance with legal standards for truthfulness in advertising, ensuring that users can rely on the integrity of these claims.

Selection Procedure. We aimed to create a corpus of secure messaging apps that made privacy claims to users, such that it included widely-used apps and was of a tractable size to perform our analyses. To create this corpus, we first had to identify a set of the most popular secure messaging apps in the Google Play Store. We focused on apps in the `Communication` category in the Google Play Store, which included a broad range of messaging apps, including email clients, mobile browsers, and SMS apps. Within this category, we used open-source tooling[7] to identify apps whose descriptions included one or more keywords related to online messaging[8] and explicitly excluded keywords related to non-messaging apps.[9]

To establish this list of keywords, we manually reviewed the descriptions of apps in the `Communication` category and iteratively added keywords to our inclusion and exclusion lists until we manually determined that the resulting set of apps included secure messaging apps that do not fall back onto SMS. Then, we excluded any app whose description did not include the terms "privacy" or "security." Finally, we only selected apps with more than a million installations. This penultimate set contained 24 apps. We decided not to analyze Google

[6] 15 U.S.C. §45.

[7] https://github.com/facundoolano/google-play-scraper

[8] "messaging," "chat," "internet," "friend," and "in touch."

[9] "SMS," "browser," "VPN," "recover," and "voicemail."

Messages because it is owned by Google and, therefore, there is no notion of third-party leakage in that app; Google runs the infrastructure that provides the push notifications. We also excluded Leo Messenger, which appeared to aggregate other messaging apps and did not have messaging functionality in its own right, as well as Gap Messenger, for which we were unable to register. Therefore, the final set contained 21 apps.

App Analysis

We performed dynamic and static analysis on each secure messaging app in our data set to learn how the usage of FCM impacted user privacy. Specifically, did the app naïvely leverage the default FCM behavior and include plaintext user content? Or, did the app use specific techniques to protect the privacy of user messages above and beyond what FCM offers by default? (For example, by integrating the Capillary library [27] mentioned in Google's blog.)

Data Types. In our analysis, we searched for specific data types that we expected to appear in the content of push notifications. To compile the list of these data types, we started with the data types defined and used by Google's privacy labels [83], which also enabled us to compare observed practices with the privacy labels declared by each app's developer. As we present in Section 5.5, we found evidence of the following data types being leaked to Google: (1) *Device or other IDs*, (2) *User IDs*, (3) *Name*, (4) *Phone Number*, and (5) *Message Contents*. Unlike (1) to (4), the contents of communications are afforded additional protections in many jurisdictions due to their sensitive nature.[10] We present additional information about these data types in Appendix B.1.

We performed our analysis in early 2023 with an instrumented version of Android 12, at a time when the majority of users (more than 85%) had Android version 12 or below installed on their phones [186]. Using a Pixel 3a phone, we installed each app from Google Play Store and saved its Android package (APK) files and privacy disclosures. We also created test accounts where necessary. We then used dynamic analysis to identify what personal information got leaked to FCM and static analysis to understand what strategies apps used to protect user privacy.

Data Leakages. We used dynamic analysis to record the contents of a push notification after our device received it from the FCM server. We instrumented the `keySet()` method of the standard `BaseBundle` class [80], which gets called by the FCM SDK, and logged the contents of the `Bundle` only if it contained the default keys in a push notification, such as "google.message_id." Additionally, we used Frida [75] to instrument the `handleIntent` method of `FirebaseMessagingService` [82], which listens and receives FCM push notifications as broadcasts from Google Play Services. This method then delivers push notification contents to app-specific callback methods (e.g., `onMessageReceived`), which allow the app to handle and display push messages as notifications to users.

[10] E.g., Title I of the Electronic Communications Privacy Act of 1986 (ECPA) [204].

The main goal was to trigger a push notification so that the resulting payload sent from Google's FCM server to our test device could be recorded (connection 3 in Figure 5.2). We installed each app on two devices and triggered push notifications by sending messages from one device to another. On the recipient's Pixel 3a device, we recorded the push notification contents as they were received by the app using the instrumented methods.

Privacy Strategy. The push notifications that we observed fell into one of the following three categories:

1. **No Protection.** The FCM push notification contained all of the information (i.e., username and message contents) that the app uses to display the notification.
2. **Some Protection.** The FCM push notification contained some personal information but, notably, did not include the displayed message contents in plaintext.
3. **Full Protection.** The FCM push notification did not contain any personal information, and any additional fields were empty or always contained unique values (i.e., not corresponding to any persistent identifiers).

For the first case, we simply assumed that the app does not use any privacy protection strategies. For the latter two cases, determining the strategy was often straightforward. For instance, Skype (in secret chat) included `EndToEndEncryption` as the value for the `messagetype` key, while Session included the `ENCRYPTED_DATA` key with a value corresponding to an encoded message. Signal, on the other hand, received FCM push notifications that only contain the empty field `notification` without any other content.

To validate the identified strategies, we performed static analysis. We first decompiled the APKs for each closed-source app using the `jadx`[11] Dex to Java decompiler. Analyzing obfuscated code was often complex. We searched for `FirebaseMessagingService` to find services that extend it. We then examined the code of these services to see how they implement the `onMessageReceived` method, which gets invoked by the FCM SDK whenever the app running on the client device receives a push notification. Crucially, the SDK also passes a hash table of type `RemoteObject` containing information necessary to display the notification to the user and, optionally, a data payload to perform any custom functions triggered by the receipt of a notification.

We tried to determine whether the push notifications contain sensitive content by observing the strings defined in code and used in the names of the keys or in print statements. We then traced the message and any variables assigned to the sensitive content until we reached the code for displaying the notification to the user. Appendix B.2 includes the questions we used to analyze the source code of apps in our data set.

[11] https://github.com/skylot/jadx

App	Privacy Strategy	Message Content	Device IDs	User IDs	Name	Phone #
Skype (default)	None	●	●	●	●	○
(secret chat)	E2EE	○	●	●	●	○
Snapchat	E2EE	○	○	●	●	○
Viber	Push-to-Sync	○	●	●	○	●
LINE	E2EE	○	○	○	●	○
Discord	None	●	○	●	●	○
WeChat	None	●	○	●	●	○
JusTalk	None	●	○	●	●	○
SafeUM	E2EE	○	○	●	○	○
YallaChat	E2EE	○	○	●	●	○
Comera	Push-to-Sync	○	○	●	○	●
Wire	Push-to-Sync	○	●	●	○	○

Table 5.1: List of apps leaking personal information to FCM servers.

Privacy Disclosure Analysis

The final phase of our analysis involved comparing the claims that app developers made in their privacy disclosures to the ground truth that we observed from our dynamic and static analysis. Therefore, we focused on the 11 app developers that we observed including personal information in the push notifications sent via Google's FCM (§ 5.5). We wanted to determine whether a user could reasonably conclude that the app guarantees the security and privacy of their personal information based on the information presented by the app vendor in their Play Store description, official website, marketing and promotional materials, and other documentation. Moreover, we wanted to understand whether developers disclose the sharing of personal information for the purposes of providing push notifications in their privacy policies.

To achieve this, several researchers from our team first located statements by app vendors that talk about the security and privacy of messages. We also determined whether the apps (that we observed sharing personal information with Google) claimed to support end-to-end encryption by default, potentially misleading the users about the privacy of their messages or their metadata. Finally, we read each privacy policy to determine whether they stated that the particular types of personal information we observed might be shared with service providers for the purpose of app functionality. If it did, we further recorded whether the privacy policy listed the specific service providers or the specific types of data shared for the purpose of app functionality, which we compared against the results of our app analysis. By cross-referencing the different sources of information about an app's privacy practices, we aimed to build a holistic picture of how each developer frames the privacy risks associated with use of their app. We saved static copies of each privacy disclosure and the privacy policies using the Internet Archive's Wayback Machine [16].

Ethical Research

Our work involves reverse-engineering the client apps of popular Android secure instant messengers in order to glean the types of information being leaked to Google's FCM servers in push notifications. We performed our analysis by running each app on our test devices, with test accounts, on a segmented and private network, and observing both the network traffic that resulted and, when that network traffic did not reveal personal information, the static code. We were only interested in observing the leakage of personal information pertaining to our test devices; we did not interact with other app users nor did we make any attempts to obtain personal information of other users. Our study did not involve human subjects, nor did it involve unauthorized access to protected systems.

As we discuss in Section 5.5, we found inconsistencies between the observed app behavior and promises made by developers of several apps from our data set (see also Table 5.1). We disclosed our findings to those developers to ensure these inconsistencies can be addressed promptly (see § 5.7 for a further discussion).

5.5 Results

We present findings from our analysis of secure messaging apps, including the personal information observed being shared with Google's FCM servers and the mitigation strategies employed by apps to prevent such leakage. Additionally, we analyzed statements made by app developers to determine whether they make any privacy or security guarantees and whether they disclose the sharing of personal information for push notifications.[12]

App Analysis

We found that almost all analyzed applications used FCM. Of the popular secure messaging apps that we identified, 20 of 21 apps relied on FCM to deliver push notifications to users. One exception among those apps was Briar messenger, which prompted the user to enable unrestricted battery usage, allowing the app to poll for new messages in the background. (Several other apps in our dataset also prompted us to enable unrestricted battery usage, however, those apps still relied on FCM.) Since our study focuses on FCM, we excluded Briar and analyzed only those applications that relied on FCM to deliver push notifications.

Of the 20 apps we analyzed, 11 included personal information in data sent to Google via FCM such that that data was visible to Google. All 11 apps leaked message metadata, including device and app identifiers (3 apps), user identifiers (10 apps), the sender's or recipient's name (7 apps), and phone numbers (2 apps). More alarmingly, we observed 4 apps—which have cumulative installs in excess of one billion—leak message contents. We present information about the observed practices in Table 5.1.

[12]Supplemental materials are available at https://github.com/blues-lab/fcm-app-analysis-public.

App	Version	Uses FCM?	Privacy Strategy	Observed Data Leakage	Min Installs (millions)
Facebook Messenger	v403.1.0.17.106	●	e2ee	□	5,000
WhatsApp	v2.23.12.78	●	Push-to-Sync	□	5,000
Skype	v8.93.0.408	●	**none** (default) e2ee (secret chat)	⊠	1,000
Snapchat	v12.28.0.22	●	e2ee	⊠	1,000
Telegram	v9.4.4	●	e2ee	□	1,000
Viber	v19.4.0.0	●	Push-to-Sync	⊠	1,000
LINE	v13.4.2	●	e2ee	⊠	500
Discord	v172.24	●	**none**	⊠	100
Kakao Talk	v10.0.7	●	e2ee	□	100
Kik	v15.50.1.27996	●	Push-to-Sync	□	100
Signal	v6.11.7	●	Push-to-Sync	□	100
WeChat	v8.0.30	●	**none**	⊠	100
JusTalk	v8.6.10	●	**none**	⊠	10
SafeUM	v1.1.0.1548	●	e2ee	⊠	5
YallaChat	v1.4.2	●	e2ee	⊠	5
Briar	v1.4.23	○	Polling	□	1
Comera	v4.0.1	●	Push-to-Sync	⊠	1
Element	v1.5.22	●	Push-to-Sync	□	1
Session	v1.16.7	●	e2ee	□	1
Threema	v5.0.6	●	Push-to-Sync	□	1
Wire	v3.82.38	●	Push-to-Sync	⊠	1
				TOTAL installs	**15,026**

Table 5.2: The complete dataset of analyzed secure messaging apps.

It is worth noting that not all of the observed behaviors here are necessarily *undisclosed sharing*. Undisclosed sharing occurs when data we observed being shared from our static and/or dynamic analysis was not disclosed in the privacy disclosures we analyzed. Whether the observed behaviors do constitute undisclosed sharing depends on the findings from our privacy disclosure analysis, discussed below (§5.5).

Mitigation Strategies

Of the 16 apps that did not send message contents to Google.[13] our static analysis revealed two general mitigation strategies described below: end-to-end encryption and push-to-sync. Ultimately, we observed 9 apps out of 16 employ either end-to-end encryption or push-to-sync strategies to prevent leaking any personal information to Google via FCM. The remaining 7 apps still leaked metadata, but not the message contents. See Table 5.2 for more information.

[13]Skype used e2e encryption to protect message contents only in secret chats, which is not the default.

```
firebase:message:10276:START:{
    google.delivered_priority=high,
    google.sent_time=1677001395829,
    google.ttl=2419200,
    google.original_priority=high,
    from=312334754206,
    google.message_id=0:1677001395846147…,
    notification=,
    google.c.sender.id=312334754206
}
```

Figure 5.4: Example payload inside the `RemoteMessage` received by Signal.

End-to-End Encryption. We determined that 8 apps employed an end-to-end encryption strategy to prevent privacy leakage to Google via FCM. In this strategy, when the user launches the app for the first time, the app provisions a keypair and does a secure key exchange between the user's device and the app's server. The app will then develop a session key that it can use to decrypt messages from the server. The server encrypts messages it sends using the session key before it goes to FCM.

As depicted in Table 5.2, of the 8 apps that utilized the end-to-end encryption (e2e) strategy, only 4 (Facebook Messenger, Telegram, Session, and KakaoTalk) did not leak *any* personal information to Google via FCM. The remaining 4 (Snapchat, SafeUM, YallaChat, and LINE) still leaked metadata, including user identifiers (3 apps) and names (3 apps).

Push-to-Sync. We observed 8 apps employ a push-to-sync strategy to prevent privacy leakage to Google via FCM. In this mitigation strategy, apps send an empty (or almost empty) push notification to FCM. Some apps, such as Signal, send a push notification with no data (aside from the fields that Google sets; see Figure 5.4). Other apps may send an identifier (including, in some cases, a phone number). This push notification tells the app to query the app server for data, the data is retrieved securely by the app, and then a push notification is populated on the client side with the unencrypted data. In these cases, the only metadata that FCM receives is that the user received some message or messages, and when that push notification was issued. Achieving this requires sending an additional network request to the app server to fetch the data and keeping track of identifiers used to correlate the push notification received on the user device with the message on the app server.

As detailed in Table 5.2, only 5 (Whatsapp, Signal, Threema, Element, and Kik) did not leak any personal information to Google. The remaining 3 (Viber, Wire, and Comera) leaked metadata, including user identifiers (all 3 apps), device and app identifiers (2 apps), and phone numbers (2 apps).

```
firebase:message:10279:START:{
    google.delivered_priority=high,
    google.sent_time=1677010922128,
    google.ttl=2419200,
    google.original_priority=high,
    resend=0,
    MtcImTextKey=Hello Dustin! How are you doing?,
    MtcImTimeKey=1677010922031,
    MtcImUserDataKey={},
    MtcImInfoTypeKey=Text,
    from=144552557193,
    toUid=9999_43035938,
    google.message_id=0:1677010922135234%...,
    MtcImLabelKey=P2P/9999_43036012,
    MtcImDisplayNameKey=Charlotte,
    google.c.sender.id=144552557193,
    MtcImMsgIdKey=0,
    MtcImImdnIdKey=97866160-0e6a-495a-9932...,
    MtcImSenderUidKey=9999_43036012
}
```

Figure 5.5: Example payload inside the `RemoteMessage` received by JusTalk.

Privacy Disclosure Analysis

We analyzed privacy disclosures for the 11 apps that included personal information in the push notifications sent via Google's FCM. One of our aims was to determine whether a user could reasonably conclude that the app guarantees the security and privacy of their personal information based on the information presented by the app vendor in their Play Store description, official website, marketing and promotional materials, and other documentation. Table 5.3 provides details for each app.

Marketing Claims. First, we discuss the 4 apps that leaked the actual contents of push notification messages: Skype, WeChat, Discord, and JusTalk. We found that out of these four apps, only JusTalk claimed to be end-to-end secure, stating: "All users' personal information (including calling and messaging data) is end-to-end encrypted and is split into multiple random paths which ensure it can't be monitored or saved by servers. Moreover, all the personal data is never shared with any third party. Enjoy safe and free calls" [109]. Nevertheless, we clearly observed the contents of our messages being sent without end-to-end encryption via FCM's servers while delivering push notifications (see Figure 5.5).

Although the three remaining apps do not claim to employ end-to-end encryption, both WeChat and Discord made statements about their concern for privacy. For instance, WeChat said in their Play Store description: "- BETTER PRIVACY: Giving you the highest level of

App		E2EE	S/P	Discloses PI Sharing	Discloses Companies	Discloses Shared PI
Skype	(default)	○	○	●	●	○
	(secret chat)	●	●	●	●	○
Snapchat		○	●	●	●	○
Viber		●	●	●	○	○
LINE		●	●	●	●	○
Discord		○	●	●	●	○
WeChat		○	●	●	○	○
JusTalk		●	●	●	●	●
SafeUM		●	●	●	●	○
YallaChat		●	●	●	●	●
Comera		●	●	●	○	○
Wire		●	●	●	●	○

Table 5.3: Disclosures connected to apps leaking personal information to FCM.

control over your privacy, WeChat is certified by TRUSTe" [155]. Although Skype does not reference secure messaging for their normal (default) chat functionality, they promise that "Skype private conversations uses the industry standard Signal Protocol, allowing you to have end-to-end encrypted Skype audio calls, send text messages, image, audio, and video files" [137]. Although we did not observe the content of the message being leaked when testing Skype's private conversation feature, we still observed the app leaking device IDs, user IDs, and names via Google's FCM.

For the remaining 7 apps that did not leak message contents, we observed each of these apps make claims that could lead users to believe that the apps do not share any personal information with anyone and, except for Snapchat, claimed to be end-to-end encrypted. For instance, SafeUM messenger put it plainly: "[w]e never share your data with anyone. Never" [165].

Privacy Policies. We additionally read each privacy policy to understand whether developers disclosed the sharing of personal information for the purposes of providing push notifications. We found that all 11 apps that shared personal information with Google's FCM servers stated that personal user data may be shared with service providers (such as FCM) for the purpose of app functionality. However, only two apps (JusTalk and YallaChat) enumerated the types of personal information shared with such service providers, which did not cover the types of information we observed being shared, namely user IDs and names (for both apps) and message contents (for JusTalk, as discussed above). Furthermore, three apps (Viber, WeChat and Comera) did not specify which companies serve as their service providers. Out of the remaining 8 apps, only 4 mentioned Google in the context of push

notifications and/or FCM.

Given that only YallaChat included information about the types of data shared with Google's FCM, we were unable to determine whether the specific data types we observed being shared would be covered by these statements or not. For instance, Viber's privacy policy stated, without giving any specifics: "[w]e may disclose your Personal Information to a contractor or service provider for a business purpose. The types of personal information we share for a business purpose, vary, depending on the purpose and the function provided by the third party to whom we disclose such information" [214]. While these statements may technically address personal data sharing in the context of push notifications, they do not meaningfully inform users about *what* information pertaining to them is being shared and *with whom*.

5.6 Discussion

The democratization of mass communications via the Internet has created a new paradigm in which anyone can have a platform to send a message. Consequently, anyone can now become a software engineer and distribute software worldwide. By and large, this is a good thing. However, it raises issues of professional responsibility that have long been addressed by other more mature branches of engineering. In most jurisdictions, one cannot simply decide to become a civil engineer and erect a multi-story building. Due to the inherent safety risks—to the individual, neighbors, and society—most jurisdictions require that plans be submitted for approval. In granting that approval, the plans are first checked for conformance with building codes, which have been set (and periodically revised) by professional societies with deep expertise. Once plans are approved, multiple levels of oversight still occur: at various steps during construction, building inspectors confirm that both the plans have been followed and that no other safety issues have been identified. Moreover, after construction has been completed, governments are empowered to continually monitor for code violations, going so far as to condemn structures that pose safety hazards. Of course, there is a reason for this oversight: building codes are written in blood.

In the past decade or two, software engineering as a discipline has only just begun to reckon with the complex sociotechnical issues relating to harm and liability. While the collapse of a building is likely to be more lethal than the inappropriate exfiltration of sensitive user information, the latter may still pose risks to user safety—even lethal ones. We chose to examine secure messaging apps in this study because they can often embody these risks: online messaging apps are increasingly being used by activists living in oppressive regimes [203], who may find themselves in serious jeopardy if their communications are inappropriately revealed. In this specific instance, the inappropriate disclosure of users' communication and metadata does not require malice on the part of a service provider for harm to come to the user. By nature of such data collection, the service provider exposes the user to legal processes: this may result in data the user legitimately did not believe to exist coming into

the hands of governments and private actors. We emphasize that this risk is not merely theoretical; as previously noted, U.S. Senator Ron Wyden published a letter that confirms that government agencies do, in fact, collect user information by demanding push notification records from Google and Apple [228].

Our analysis found that several prevalent secure messaging apps—which imply that they will not share certain information with third parties—do indeed share that information in plaintext with Google via FCM (see Table 5.1). We found evidence of undisclosed data leakage to FCM in apps that account for over 2 billion installs. Users of these apps are likely unaware of these data leakages: some of the privacy disclosures made by these apps often explicitly promise *not* to share such personal information with third parties, whereas others were so vaguely written that it was unclear whether these behaviors are being disclosed (and how they might comport in consumers' minds with the companies' marketing materials that imply messaging data will be kept private). Consequently, consumers may have a false sense of security when using these apps for communicating. The undisclosed leakage of communication contents can harm users and potentially even innocent bystanders who may be mentioned in communications.

Recommendations

Just as a contractor or owner-builder is ultimately responsible for the adherence to local building codes and the risks associated with deviations from them, software developers publishing apps for public usage are responsible for the behaviors of those apps. This responsibility includes verifying that third-party components function as expected and that the ultimate behavior of the app is in accordance with platform guidelines, the developer's disclosures, and applicable laws/regulations. The use of these third-party components is not unique to software engineering: other branches of engineering generally involve complex supply chains, yet there is often a great deal of oversight. When Airbus builds a plane, they may use engines from Rolls-Royce or electronics from Siemens; but in addition to simply specifying the specifications and tolerances that Airbus expects these components to conform to, they nonetheless validate those third-party components by launching chickens at them at 600+ km/h (amongst other validation tests) [225]. Such integration validations rarely exist for software *in practice*, despite being recommended for nearly half a century now [74]. Indeed, while we have no reason to believe that misleading or confusing security and privacy claims are the result of malice, we believe that the poor privacy practices that we document in this paper could have been discovered and mitigated by the developers had they inspected the traffic sent and received by their applications during quality assurance processes. Thus, we offer recommendations to different stakeholders on ways to address the identified security and privacy issues.

App Developers

As the parties ultimately responsible for their apps, app developers should perform the type of dynamic analysis that we performed in this study as part of each and every release cycle. This will help to ensure that users' personal data flows in accordance with reasonable expectations, applicable laws and regulations, as well as platform policies. However, the best way to ensure that push notifications do not leak sensitive user information is to avoid sending sensitive user information via FCM in the first place. We argue that developers should implement the push-to-sync approach: the developer's server should only send the app a unique notification ID via FCM, which can then be used to fetch the notification content from the developer's servers securely. Several developers correctly used the push-to-sync approach, which resulted in no personal data being leaked by those apps. Others should adopt this architecture in their apps.

Platforms and SDK Providers

At the same time, platform owners and SDK providers are well-positioned to identify and correct issues in their tools and highlight security and privacy risks in their documentation. For its part, Google provides an API that results in developers systematically making very similar privacy mistakes. This is not helped by Google's guidance, which instructs developers to "send as much data as possible in the FCM payload," and that if they want to do so securely, they must use an additional library [172]. This guidance departs from Google's own data minimization and secure-by-default principles [81] and recommendations from other push notification providers, such as Apple [15].

We argue that the availability of usable, secure push notifications libraries, including Google's Capillary [27], does not solve the underlying problem. Developers generally trust Google's security practices and are largely unaware of the risk of personal information leakage via push notifications. Furthermore, under current regulatory regimes, Google is not obligated to do anything about this: they provide a free API for developers, and despite the fact that using it to send messages *securely* admittedly takes additional non-obvious steps, there are no legal requirements that Google—or any other SDK provider—provide a secure-by-default API. Furthermore, as mentioned previously, Android app developers are effectively required to use Google's FCM to send push notifications for battery consumption reasons. We argue, therefore, that real-world change will require either applying regulatory pressure or other market-corrective forces on platform owners to enforce privacy-by-design principles for critical SDKs in the software supply chain, such as Google's FCM. Such a change would improve the privacy and security of nearly all Android apps, because the use of FCM to deliver push notifications on Android is nearly universal.

The use of these types of APIs also represents the classic usable security problem (wherein application developers are the "user"): the user is not qualified to be making the decisions that are forced upon them, whereas those forcing them to make these decisions are in a much better position to make those decisions on the users' behalf. Prior research shows

that developers, despite being the party ultimately responsible for the behaviors of their software, are woefully unprepared to make these types of decisions [9, 2]. And thus, we are faced with a situation in which the parties most equipped to fix these types of problems (e.g., by creating more usable documentation that highlights security and privacy risks, making SDK settings secure by default, proactively auditing how their services are used in practice, etc.) are not incentivized to do so, whereas the parties who are ultimately responsible are generally incapable and do not understand their risks or responsibilities. As a result, this is fundamentally an economics problem concerning misaligned incentives [10]: in a perfect world, the responsibility for handling users' data responsibly would be placed upon those according to their abilities, shifted from those according to their needs [133]. This is not the world in which we currently live.

Yet, things are improving. In recent years, the U.S. Government has promoted the strategy of shifting the burden of software security away from individuals, small businesses, and local governments and onto the organizations that are most capable and best-positioned to reduce risks [196]. In line with this initiative, the U.S. Cybersecurity and Infrastructure Security Agency (CISA) and 17 U.S. and international partners published an update in August 2023 to joint guidance for implementing secure-by-design principles [56]. One secure product development practice, in particular, highlights the need to provide secure defaults for developers by "providing safe building blocks...known as 'paved roads' or 'well-lit paths.'" We believe that push notification providers can similarly apply privacy-by-design principles [149] to safeguard the privacy of users who cannot easily manage the risks.

Without correctly aligned incentives to motivate platforms and SDK providers to make their systems secure by default (including documentation that highlights security and privacy risks), developers will continue to be placed in this position and will continue to consistently make these types of mistakes. Thus, until software engineering becomes a more mature field with formalized oversight, validation, disclosure, and auditing procedures, these types of errors will proliferate, leaving end users at risk.

5.7 Responsible disclosure

Responsible disclosure is a critical component of security and privacy research. We reported our substantive findings to the 11 app developers who leaked at least one personal data type to Google's FCM service. We tried contacting the developers via various contact methods, including formal bug bounty programs, emailing security teams, or failing that, general support contacts. The app developers for whom we could find contact information were sent a summary report on or before June 7, 2024. We received an acknowledgment of our email from 5 developers of the 11 we contacted.

At the time of publication, the remaining 6 app developers to whom we disclosed our findings had not replied; discussions are ongoing with several companies regarding how they should fix the identified issues. We look forward to continue engaging in productive

conversations to help developers understand how to adapt their push message architectures to better protect user privacy.

5.8 Limitations

Many apps beyond secure messaging apps might send private data through push notifications. Our study only focused on secure messaging apps because most of them claim to focus on user privacy, thus, they would be among the most likely apps to take proactive steps to prevent the leakage of user data to FCM (and presumably users of these apps are more likely to believe that their communications are secure). We suspect that privacy leakage via Google FCM may be even more prevalent within apps in other contexts. Future work should look at both less popular secure messaging apps and apps in other contexts to observe to what extent, if at any, they mitigate the leakage of sensitive personal data to Google via FCM.

We also performed our analysis using an older Pixel 3a device running Android 12. We are unaware of any substantial changes in Android 13 and 14 that would have a material impact on our observed findings. Our device supported security updates and the installation of all the apps that we analyzed for this research. We ran these apps and received push notifications from FCM without observing any undesirable impact on app performance. Furthermore, at the time we began our analysis in early 2023, the majority of users (more than 85%) used Android version 12 or below [186]. While most people who use a mobile phone use an Android device, iOS also has a significant share of the mobile phone market and tends to bill itself as having more privacy-preserving practices. Future work can also explore whether private user data is leaked to Apple or other third parties via the push notification infrastructure available to developers in the iOS ecosystem.

We looked specifically at privacy leakage through push notifications that rely on FCM. As far as we know, FCM is also used in other applications, on Android and beyond; how this fact affects privacy leakage across other applications is not well understood. Future work could investigate the privacy implications of FCM across those applications. Within the Android ecosystem, there may exist other patterns or tools provided by Google or by other popular third-party libraries that also incur unexpected privacy leakage. Future work could look for such patterns beyond the Android platform, such as iOS, and identify how other ecosystem players like Apple and Google can craft a more trustworthy ecosystem to provide more privacy-preserving defaults to the broadest base of users.

Chapter 6

From Principles to Practice: A Case for Privacy Engineering

This chapter examines why developers struggle to implement robust privacy protections despite established frameworks and principles. It highlights the economic and organizational pressures that prioritize data-driven functionalities over privacy safeguards, as well as the scarcity of straightforward privacy-enhancing tools. Cultural factors within companies—such as a focus on quick releases—often overshadow privacy-by-design efforts. Moreover, individual developers frequently lack the necessary training, viewing privacy as secondary or equating it solely with security, while legal and policy guidance fails to translate smoothly into technical requirements. Academic solutions often remain impractical within modern software development processes. These challenges suggest a need for specialized professionals—privacy engineers—who can bridge the gap between principles and practice, offering technical expertise, guidance, and advocacy. Though not a complete solution, privacy engineers represent a promising new role that could help align organizational priorities, tools, and methods to ensure privacy protection is integrated throughout the software development lifecycle.

Given the privacy failures discussed in Chapters 4 and 5 of this thesis, and the significant body of prior work documenting similar shortcomings, we are compelled to ask a crucial question: why do developers struggle to implement privacy protection principles into their software? More specifically, why do many applications continue to collect, share, or process personal information in ways that deviate from declared and expected practices? These questions are not merely theoretical; they speak to the heart of a persistent disconnect between privacy principles and their practical instantiation in software systems.

This chapter explores a range of factors that hinder the effective implementation of robust privacy protections. By reviewing the key findings from existing empirical studies, it highlights the structural and contextual issues that render privacy considerations challenging for software developers. These insights help pave the way for the subsequent chapter

(Chapter 7), which introduces privacy engineers as a new class of specialists who may be particularly well-suited to bridging the gap between theoretical privacy principles and their technical realization.

6.1 Economic incentives and the privacy-utility trade-off

One of the primary forces shaping privacy decisions in software development lies in the underlying market and economic realities of the software ecosystem [3]. Implementing—and continually maintaining—effective privacy protections often entails costs that small and medium-sized development teams may find difficult to shoulder. Usable privacy-enhancing tools (PETs) that integrate seamlessly into existing workflows are scarce. Instead, developers commonly need to tailor complex solutions to their particular use cases, a process that requires specialized expertise, time, and resources [117].

More fundamentally, the perceived conflict between privacy protection and data-driven functionalities often defines a core tension within product development. Developers regularly confront a *privacy-utility trade-off*, in which privacy safeguards can seem to limit the potential business value of collecting and analyzing user data [183, 64]. From the perspective of a software company, user data can fuel targeted advertising, recommendation algorithms, and metrics for strategic product decisions. For users, these data-driven features often translate into more personalized services and improved user experiences. Thus, when developers must choose between delivering data-intensive functionalities and sustaining stringent privacy protections, the scales often tip in favor of utility—especially under conditions of tight deadlines, competitive pressures, and stakeholder demands [182].

Empirical studies have found that developers tend to prioritize core functionalities and stakeholder requirements over privacy, especially when these appear to be at odds with one another [170]. Under these circumstances, privacy-by-design principles are regularly sidelined in favor of short-term gains. The result is a persistent misalignment between privacy ideals and actual development practices.

6.2 Organizational influence and culture

Beyond economic factors, organizational cultures and incentive structures significantly influence how developers address privacy. Technology companies frequently value rapid iteration and the ethos of "move fast and break things," an approach that champions innovation and speed over caution and due diligence. Within this culture, developers can struggle to advocate for slower, more deliberative approaches to privacy [93]. Internal pressures and performance metrics often emphasize timely feature releases over the careful integration of

privacy measures, leaving developers with little structural support to champion these initiatives [189].

The absence of senior-level privacy expertise exacerbates this problem. Without clear directives or the presence of leaders who prioritize privacy, developers must make ad-hoc judgments, often lacking the necessary guidance or institutional backing [181]. Organizational dynamics thus play a crucial role in shaping how privacy is (or is not) integrated into software development. Privacy remains underrepresented in strategic decision-making, and, as a result, the subtle complexities of data protection tend to be overlooked or addressed too late in the development cycle [184].

6.3 Developers' privacy attitudes and expertise

On an individual level, developers' personal attitudes, perceptions, and knowledge directly influence how well privacy principles are integrated into software [93]. For some developers, privacy may appear as an abstract, secondary concern, overshadowed by other pressing demands such as performance optimization, feature development, or refactoring [17]. Simplistic conceptions—such as equating privacy solely with data security or encryption—fail to capture the broader social, regulatory, and ethical dimensions of privacy protection. In these cases, technical solutions like encryption, while beneficial, are insufficient to ensure that systems respect data minimization, informed consent, and purpose limitation [182].

Lack of privacy-specific training and educational resources further compounds these challenges. Many developers enter the profession with little to no exposure to established privacy frameworks (e.g., Fair Information Practice Principles, Privacy by Design, Data Minimization) or regulatory requirements [170]. While privacy regulations continue to evolve, developers often lack guidance on how to translate high-level legal and policy mandates into concrete technical requirements [157]. In some organizations, developers may assume that privacy is the exclusive purview of legal or compliance teams, which fragments accountability and leads to ad-hoc or inconsistent approaches to privacy throughout the software's lifecycle [93].

6.4 Misalignment between theory and practice

Efforts by academia, standards bodies, and regulators to promote privacy-aware development do not always align well with the realities of contemporary software engineering. Many academic approaches, though conceptually sound, remain difficult to integrate with agile methodologies, continuous integration pipelines, and microservice-based architectures [89]. Proposed frameworks and tools often lack the "off-the-shelf" compatibility that developers need [149, 9]. As a result, even well-intentioned developers find themselves unable to operationalize privacy solutions that were designed in theoretical or laboratory settings [90].

This misalignment is further amplified by communication gaps between different stakeholders [24]. Policymakers and legal scholars often frame their guidance in terms of abstract

principles or compliance requirements. Developers and engineers, on the other hand, think primarily in terms of architecture, code, and implementation details. Absent a clear and systematic translation process, these two worlds fail to connect [26]. Privacy risks and requirements identified in policy documents remain disconnected from the daily tasks and decision-making processes of software teams [9].

6.5 Making a case for the privacy engineering profession

The persistent difficulties outlined above—from economic pressures and organizational cultures to developers' knowledge gaps and the misalignment of theory with practice—demonstrate why so many teams struggle to implement privacy frameworks and principles. Although various stakeholders (developers, legal professionals, product managers) share responsibility for privacy, this diffuse accountability often leads to incomplete solutions and recurring lapses.

Prior research examining software developers' experiences with privacy has consistently highlighted several key challenges. Studies have shown that developers struggle with interpreting privacy requirements, lack adequate tools and guidelines for privacy implementation, and face difficulties balancing privacy considerations against other development priorities. These findings suggest a fundamental gap in expertise and support for privacy implementation in software development.

This situation poses a fundamental question: who is best positioned to ensure robust privacy protection in software systems? Generalist software developers and architects have the technical skill to implement complex features but may not have the bandwidth or training to incorporate privacy thoroughly. Legal teams understand the regulatory landscape but often lack the ability to translate rules into system-level requirements. Organizational leadership sets priorities but may not appreciate the technical and logistical challenges that privacy integration entails.

These gaps suggest the need for a new role dedicated to bridging these domains and ensuring privacy is woven into every stage of the Software Development Lifecycle (SDLC). *Privacy engineers* have emerged as such a specialized profession, bringing together deep technical expertise, familiarity with privacy frameworks, and an ability to communicate and coordinate across legal, managerial, and engineering domains.

Privacy engineers differ from lawyers and compliance officers who focus on regulatory adherence rather than on system-level architectural decisions. They also stand apart from security engineers who, while often related in skillset, concentrate on preventing malicious intrusions rather than ensuring that the system's inherent data practices align with privacy principles. By blending technical acumen and an understanding of privacy theory and policy, privacy engineers can guide architectural decisions, identify risks, and apply privacy-preserving patterns at an early stage in the development process.

6.6 The promise and limitations of privacy engineers

While the emergence of privacy engineering as a distinct profession holds promise, it is not a cure-all. Organizational commitment, adequate resources, and ongoing support remain crucial. Privacy engineers can identify and recommend privacy-focused solutions, but if companies reward quick feature releases over careful privacy integration, even a well-trained privacy engineer may struggle to effect meaningful change.

Additionally, the field itself is new and evolving. Training pipelines, best practices, and professional norms for privacy engineers are still taking shape. Some organizations may be hesitant to invest in or fully integrate a role that, until recently, did not exist as a well-established position in the software development hierarchy.

Nevertheless, the existence of a dedicated privacy engineering profession represents an important step forward. By acknowledging the multifaceted nature of privacy challenges—economic, organizational, individual, and structural—and positioning specialized professionals to address them, organizations can begin to close the gap between the aspiration to protect users' privacy and the realities of implementing effective, privacy-preserving systems.

In summary, this chapter has explored the myriad challenges that prevent developers from fully realizing privacy principles in software. From economic trade-offs and organizational cultures that undervalue privacy to the ongoing difficulty of translating theoretical frameworks into practical tools, the path to robust privacy protection is fraught with obstacles. These insights set the stage for Chapter 7, which examines the rise of the privacy engineering profession through an in-depth interview study. There, we will see how privacy engineers operate in the field, how they navigate the complexities outlined in this chapter, and what competencies and strategies they employ to ensure that privacy principles truly take root in contemporary software development.

Chapter 7

How do Privacy Engineers Engineer Privacy?

This chapter examines the rapidly evolving landscape of privacy protection, where complex regulations and shifting societal expectations drive a growing need for privacy engineering. As organizations grapple with new laws, record-breaking fines, and diminished consumer trust, the demand for technical privacy expertise intensifies. Privacy engineering has emerged as a critical function yet remains loosely defined and challenging to integrate into organizational structures.

To address this gap, we conducted 28 semi-structured interviews with practicing privacy engineers and analyzed 12 interview transcripts until no new themes emerged. Our results highlight wide variation in job titles, responsibilities, and skill sets, alongside persistent organizational challenges and unclear evaluation criteria. Participants described the need for technical expertise, ethical motivations, and the importance of cross-functional collaboration. These findings enrich academic and industry understanding of privacy engineering, informing better role definitions, guidance for aspiring professionals, and potential frameworks for regulators to foster clearer, more consistent privacy standards.

7.1 Introduction

The landscape of privacy protection is rapidly evolving, driven by a complex interplay of technological advancements, regulatory changes, and shifting societal expectations. Organizations today face an increasingly stringent data protection environment, with many laws and regulations across different jurisdictions. In 2023 alone, 40 US states and Puerto Rico introduced or considered at least 350 consumer privacy bills, many with conflicting requirements that significantly increase the compliance burden for organizations operating across state lines [140]. This regulatory complexity is further compounded by robust enforcement actions, as exemplified by the record-breaking €1.2 billion fine imposed on Meta in 2023

for violating EU General Data Protection Regulation (GDPR) international data transfer requirements [1].

Beyond regulatory pressures, privacy has emerged as a critical factor in maintaining consumer trust and competitive advantage. The impact of privacy concerns on consumer trust is not to be underestimated, as a survey of nearly 5,000 consumers from 19 countries revealed that 68% are concerned about online privacy, highlighting the growing importance of digital trust in shaping user expectations and behaviors [138]. This concern directly impacts business outcomes, with another survey reporting that 76% of consumers discontinued using products and buying from organizations they do not trust with their data [136]. The competitive implications of privacy practices are further illustrated by the mass exodus of millions of users from WhatsApp to Signal following a global backlash over WhatsApp's privacy practices allowing data sharing with its parent company, Meta [112].

In response to these multifaceted challenges, privacy engineering has gained significant traction in business and academia. Large enterprises are establishing dedicated privacy engineering departments, recognizing the need for specialized technical expertise in addressing privacy concerns [149]. The growing importance of privacy engineering is further underscored by its inclusion as one of four key areas in McKinsey's 2022 technology trends outlook [44].

Despite this rapid growth and recognition, privacy engineering remains a field in flux, with no universally accepted definition or clear professional boundaries. Conceptions of privacy engineering range from the narrow focus on designing and implementing anonymity-preserving algorithms and protocols to broader approaches that incorporate methods from software engineering, human-computer interaction, and socio-technical systems design. This lack of clarity poses significant challenges for organizations in defining job roles, recruiting qualified professionals, and integrating privacy engineering functions into their operations. It also creates uncertainty for individuals interested in pursuing privacy engineering as a career or transitioning from related fields.

To address this knowledge gap, we conduct in-depth interviews with privacy engineers and professionals across various industries and geographical locations. We seek to identify and analyze the different role types, responsibilities, skill sets, and experience requirements associated with privacy engineering positions. Furthermore, we will examine the reporting structures and organizational integration of privacy engineering functions, as well as investigate current practices and methodologies employed by privacy engineers in their day-to-day work.

The findings of this research will contribute to both academic and practical understanding of privacy engineering. By providing a more explicit definition of privacy engineering roles and responsibilities, organizations can better align their privacy strategies, improve recruitment processes, and enhance the integration of privacy engineering functions within their existing structures. Individuals interested in pursuing privacy engineering as a career will gain valuable insights into the skills, experience, and knowledge required to succeed in this field. This study will also contribute to the growing body of literature on privacy engi-

neering, offering empirical evidence to support or challenge existing theoretical frameworks. Moreover, a clearer definition of privacy engineering may assist regulators in developing more targeted and effective privacy protection guidelines and standards. By identifying common practices and skill sets across different organizations and sectors, this research may contribute to developing industry-wide standards for privacy engineering roles and qualifications.

7.2 Background

This section explores the evolving landscape of privacy engineering, discussing regulatory challenges, emerging frameworks, existing definitions, and its crucial role in aligning technological capabilities with privacy expectations in modern businesses.

Evolving Regulatory Landscape

Engineering privacy into information systems is challenging due to privacy's multifaceted and context-dependent nature [139, 142]. Privacy encompasses various physical, informational, and decisional dimensions, which can vary significantly across cultures, individuals, and situations. Privacy is inherently subjective, influenced by personal preferences, societal norms, and technological advancements. Moreover, privacy expectations evolve, making establishing a fixed, universally applicable definition difficult. Legal and regulatory frameworks attempt to codify privacy but often struggle to keep pace with rapidly changing technologies and social practices. This fluidity and complexity make it challenging for organizations and policymakers to create comprehensive and sustainable privacy solutions, as they must continually adapt their approaches to address emerging concerns and shifting boundaries of what constitutes private information or behavior.

Governments and regulatory bodies worldwide have introduced stringent data protection regulations in response to these challenges. Two of the most influential regulations that have catalyzed the growth of privacy engineering are the European Union's General Data Protection Regulation (GDPR) [68] and the California Consumer Privacy Act (CCPA) [123]. The GDPR, implemented in 2018, introduced far-reaching changes to data protection law, including data minimization, purpose limitation, and the right to be forgotten. It also mandated the principle of "privacy by design and by default," requiring organizations to embed privacy considerations into their products and services from the outset of the development process.

Similarly, several frameworks and principles have been developed to address the complex landscape of privacy requirements. Privacy by Design (PbD) [39] and the Fair Information Practice Principles (FIPPs) [77] are notable among these. These guidelines provide a structured approach to incorporating privacy considerations throughout the software development lifecycle. However, the practical implementation of these principles has proven challenging, with numerous instances of failure in applying PbD to real-world software applications. These principles, regulations, and changing norms around privacy have created

a pressing need for professionals who can identify and translate privacy requirements into technical specifications and implement privacy-enhancing technologies.

Defining Privacy Engineering

Despite its growing importance, privacy engineering lacks a universally accepted definition. Various professional groups and regulatory bodies have attempted to delineate the field, each emphasizing different aspects of this emerging discipline.

The International Association of Privacy Professionals (IAPP) broadly defines privacy engineering as "the technical side of privacy." According to the IAPP, privacy engineers are professionals who "ensure that privacy is built into products and services" [73]. This definition highlights the practical, hands-on nature of privacy engineering, positioning it as the bridge between abstract privacy concepts and their concrete implementation in technological systems.

The National Institute of Standards and Technology (NIST) offers a more technical perspective. NIST views privacy engineering as "a specialty discipline of systems engineering focused on achieving freedom from conditions that can create problems for individuals with unacceptable consequences that arise from the system as it processes PII" [30]. This definition underscores the systematic approach required in privacy engineering, framing it within the broader context of systems engineering and emphasizing its focus on mitigating risks associated with personal information processing.

In the European context, the European Union Agency for Cybersecurity (ENISA) focuses on "data protection engineering" as a set of technical and organizational measures to implement data protection principles [57]. This approach aligns closely with the requirements of the GDPR, emphasizing the need for technical and organizational strategies to ensure privacy.

The value of privacy engineers in the business world is directly tied to their ability to navigate the complex and often ambiguous landscape of privacy requirements. As specialists, they bring a methodical approach to identifying, interpreting, and implementing privacy measures in software development and data management processes. Their expertise becomes increasingly valuable as organizations grapple with evolving privacy regulations, heightened public awareness, and the potential reputational and financial risks associated with privacy breaches.

Privacy engineers ensure data utilization aligns with privacy rights and expectations in the current data-driven business environment. Their work not only helps organizations maintain compliance with a diverse array of global privacy laws but also contributes to building trust with customers and stakeholders. As privacy concerns continue to influence public discourse and consumer behavior, the role of privacy engineers in reconciling technical capabilities with privacy expectations becomes increasingly central to the success and sustainability of modern businesses.

7.3 Methodology

In this section, we explain the methodology of our interview study, including data collection and analysis. We chose semi-structured interviews due to the open-ended nature of our research questions. Furthermore, semi-structured interviews enabled us to probe further into participants' responses and skip questions as needed while maintaining the structure of our interview guide [91].

We conducted 28 semi-structured interviews with professionals who work in privacy engineering (henceforth "privacy engineers" for simplicity) between December 2023 and April 2024. We designed our interview protocol to address the following research questions:

- What types of industry roles do privacy engineers hold?
- What are the responsibilities and skills of privacy engineers?
- What do privacy engineers find challenging in their roles, and what strategies do they find least and most effective in overcoming these challenges?

To support generalizable and rigorous qualitative results, we analyzed 12 interview transcripts until new themes stopped emerging [42]. Our subject pool was larger than recommended as best practice by previous qualitative research. Therefore, our work can provide a foundation for future quantitative research and generalizable design recommendations [87].

We describe the process of developing our interview instruments, recruitment process, interview procedure, data analysis procedures, and limitations of our work. This study was approved by the Institutional Review Board (IRB) of the University of California, Berkeley.

Instruments

We describe the process of iteratively developing our screening survey and the interview guide.

Screening Survey. The screening interview aimed to identify participants based in the US who are working full-time as employees or consultants in a privacy engineering role. To gain a better understanding of the context in which the participants worked, we included questions about employment status, job title, years of experience, sector and area of employment, privacy and cybersecurity certifications, and membership in relevant industry associations. It also included basic demographic questions, which were optional. After conducting the pilot interviews (which we describe later), we adjusted the text of the questions and emphasized the optional nature of demographic questions.

Interview Guide. We developed our interview guide to highlight the unique characteristics of a privacy engineering role and to enable comparison with other similar but distinct

privacy roles. To achieve this goal, we divided the interview into six distinct sections that cover 1) participants' understanding of privacy engineering, 2) their motivation to pursue privacy engineering as a profession, 3) responsibilities and skills, 4) reporting and deliverables, 5) challenges and strategies, and 6) evaluating success. All authors reviewed and provided suggestions for the interview protocol, including two authors with extensive experience as senior privacy engineers in their respective organizations.

Pilot Interviews. Before recruiting participants for the main study, we performed three pilot interviews with privacy engineers from our personal networks. We tested and iteratively adapted both the screening survey and the interview guide after each pilot interview. These pilot interviews, while not included in the final analysis, were invaluable in validating our interview script, timing, and overall approach, leading to refinements in our interview script and improved consistency among the different interviewers.

Recruitment

The goal of our recruitment efforts was to identify privacy engineers. In our recruitment messages, we invited prospective interviewees to participate in our study if they "work in privacy engineering" or were "privacy engineers or a professional in a similar role." Therefore, we relied on participants to self-identify as privacy engineers to avoid imposing our definition of the role of a privacy engineer.

We posted recruitment messages on LinkedIn[1] to reach a broad segment of the professional privacy community members. Furthermore, we employed snowball sampling [150] and encouraged interviewees to recommend other potential participants. This method proved particularly effective in accessing privacy engineers who might not have seen our recruitment messages. The recruitment message included a link to a landing page informing candidates about the study's purpose and presenting them with our study consent form and our contact details. Participants who consented were then directed to complete our screening survey, which took an average of 4 minutes to complete. Furthermore, we asked eligible candidates to provide their email addresses to schedule a 60-minute remote interview via Zoom.

We excluded candidates who were students or worked part-time and invited eligible participants for an interview. We thanked survey respondents who did not meet our selection criteria for their interest in our research and asked them to share information about our study with other potential candidates.

Interviews

We performed semi-structured video interviews via Zoom, each lasting approximately 60 minutes, though we remained flexible to extend this time if participants had more to share. A team of four interviewers, all extensively trained in qualitative research methods and

[1] https://www.linkedin.com/feed/

privacy concepts, conducted the interviews. As discussed previously, we performed three pilot interviews to ensure consistency before beginning the main study.

Before starting each interview, we read aloud the key information on the consent form as a reminder, ensuring participants fully understood the nature of the study and how their data would be used and protected. We started the audio recording and the interview upon receiving the participants' verbal consent. The interview covered a wide range of topics, including the definition of privacy engineering, motivation and interest, required skills and responsibilities, reporting and deliverables, challenges and strategies for overcoming them, and evaluation metrics. The audio from these sessions was recorded and later transcribed.

Analysis

Our data analysis process was iterative, employing a qualitative open-coding process that allowed for the organic emergence of core concepts [42]. We divided each interview script into two parts, containing three sections each. Two pairs of researchers independently built initial codebooks based on a detailed analysis of the two interview parts. These initial codebooks were then compared by both pairs of researchers, discussed in depth, and merged, with any disagreements resolved through careful consideration and consensus-building.

The same two pairs of researchers independently coded additional transcripts to test and refine this consolidated codebook. This step led to further refinement of the codebook, with codes being added, merged, or clarified as needed to capture the nuances in the data. The final codebook was then applied to the remaining interviews, with both pairs of researchers coding all their respective interview segments to ensure consistency and reliability. Disagreements in coding were resolved through in-depth discussions between the coders.

We determined that theoretical saturation had been reached when all themes raised in new interviews fit within the existing codebook structure without requiring significant additions or modifications. At this point, we halted further recruitment and revisited the initially coded interviews, re-coding them with the final, comprehensive codebook to ensure uniform analysis across all data.

Ethics

We took great care to ensure the confidentiality of the study participants. The institutional review board (IRB) at our university approved this study. Furthermore, all participants provided informed consent before collecting any personal information. No compensation was provided to participants to avoid any potential bias. To ensure anonymity, we assigned each participant a unique code (e.g., P16) and carefully removed any identifying information from our data and results.

7.4 Results

In this section, we present an overview of participants' demographics and professional experience, as well as the themes we observed after analyzing the 12 interview transcripts.

Participants

We interviewed participants who completed our screening survey of demographics and professional questions. These questions focused on respondents' work (e.g., industry of employment, years of experience in privacy, size of organization) and demographics (e.g., age, education, income, etc.). We report demographic and professional information for the 12 participants whose transcripts we analyzed for the research in this thesis.

Demographics. The results of our survey indicated the median age of respondents was 33 years, and the majority of participants (10 or 83%) identified as men. We also found that five participants (42%) were based in California. Furthermore, most respondents (8 or 67%) held advanced degrees, including Master's and Doctorate degrees, and four participants (33%) reported earning $300,000 or more as their total annual compensation. We also found that four (33%) participants identified with communities known to be historically disadvantaged.

Professional Context. We found that all 12 participants work in the private sector, and 11 indicated working in the *Technology and Software* industry sector. Furthermore, 11 participants indicated working full-time, and half (6) reported working in large organizations with at least 100,000 people.

We also found that 10 role titles contain the word "privacy," 8 contain "privacy engineer," and five roles have some indicator of seniority (e.g., "senior" or "leader"). We also found that participants had an average of 7.5 years of professional privacy experience, working with teams of an average size of 9 individuals. More than half of the participants (9 or 75%) belong to the International Association of Privacy Professionals (IAPP)—a privacy professional association, and half (6 participants) are certified as IAPP's Certified Information Privacy Technologist (CIPT).

We now present the themes that emerged during the interviews. The current chapter provides the results for the analysis of 12 interview transcripts. Our analysis revealed four primary themes characterizing the experiences and perspectives of privacy engineers: (1) Conceptualizing privacy engineering, (2) Motivations for becoming privacy engineers, (3) Common challenges faced in their roles, and (4) Competencies and evaluation. These themes illuminate how privacy engineers understand their profession, what drives them to enter and remain in it, the obstacles they encounter, and the skills and practices they rely on to fulfill their responsibilities. Below, we present each theme along with illustrative examples from participant accounts.

Conceptualizing privacy engineering

Not strictly defined. Participants consistently emphasized that there is no universally agreed-upon definition of privacy engineering. They reported that job titles and responsibilities vary widely across organizations, even when these roles share the "privacy engineer" label. For example, some respondents described their work as deeply technical—writing code, designing systems, or conducting architectural reviews—while others considered their remit more advisory, focused on communicating policy requirements and providing internal guidance. Several participants noted that this lack of standardization led to confusion both internally and externally, with colleagues sometimes misunderstanding the scope of their role or conflating it with more familiar functions like security engineering. A few participants framed this ambiguity as a natural outcome of a relatively new field that continues to evolve, making role boundaries fluid and subject to ongoing negotiation.

Requires technical expertise. Across the interviews, participants stressed that effective privacy engineering rests on a strong technical foundation. They highlighted a need to understand complex systems, data flows, and software architectures to identify privacy risks and integrate privacy-enhancing measures. Many participants recounted instances where their ability to navigate codebases, evaluate cryptographic tools, or design novel data minimization strategies allowed them to propose practical, implementable solutions. However, participants also noted that while technical skill is critical, it must often be combined with broader privacy knowledge, including familiarity with legal frameworks and organizational policies. This combination, they suggested, is what distinguishes privacy engineers from either purely legal or purely technical professionals.

Motivations for becoming privacy engineers

Participants described a variety of reasons for entering the privacy engineering field. These motivations often reflected personal interests and values, as well as a desire for career growth and intellectual stimulation. Three themes stood out: viewing the field as novel and exciting, enjoying continuous learning and opportunities to educate others, and acting upon moral and ethical convictions related to user privacy.

Novel and exciting field. A prominent theme among participants was the allure of privacy engineering as a dynamic and emergent discipline. Participants described the role as continuously evolving, with frequent changes in technologies, regulations, and user expectations. This state of flux was seen as intellectually stimulating, driving individuals to remain current with emerging standards and tools. One participant remarked that privacy engineering never feels "boring," as each new project presents unique technical puzzles. The perceived newness of the field also gave participants a sense of pioneering work, an opportunity to shape standards and best practices in a growing domain.

Learning and teaching. Many respondents found personal fulfillment in the opportunity to learn and to teach others about privacy. Some highlighted the satisfaction they derive from gaining new skills or deepening their understanding of privacy techniques, privacy law, and human-centric design considerations. Others described their role as educators within their organizations, helping development teams comprehend privacy requirements and coaching them in implementing controls effectively. Participants noted that as they gained expertise, they became key knowledge brokers, sharing insights and fostering awareness that went beyond the day-to-day engineering tasks.

Moral and ethical values. Several participants cited moral and ethical considerations as major drivers for pursuing privacy engineering careers. They viewed privacy as a fundamental human right and expressed a sense of responsibility to protect users' data. This moral framing inspired participants to navigate organizational pressures that might deprioritize privacy considerations. In some cases, participants linked their commitment to privacy with broader ethical principles, noting that their desire to safeguard personal information was part of a larger personal ethos. Such motivations underscored a sense of purpose, setting their work apart from more purely commercial or technical undertakings.

Common challenges

While participants expressed enthusiasm for their work, they also highlighted persistent challenges. These ranged from broader organizational and market forces to difficulties translating complex regulations into actionable technical requirements. Three main themes emerged: misaligned incentives that undermine privacy efforts, the complexity of translating laws and regulations into technical standards, and structural obstacles posed by reporting and organizational hierarchies.

Misaligned incentives. A recurring challenge participants mentioned was the misalignment of incentives within their organizations. Many felt that privacy engineers are tasked with championing user data protection even when it may conflict with short-term business goals, such as quickly releasing new features or collecting more user data to enhance services. Participants shared accounts of struggling to persuade product managers or other stakeholders to invest time and resources into proactive privacy initiatives. Some participants described a reality where privacy gains visibility only after an incident or regulatory inquiry, thus incentivizing reactive rather than preventive measures.

Translating law and regulation. Participants consistently indicated that navigating the legal and regulatory environment was a complex aspect of their work. They often felt responsible for transforming high-level rules—such as those found in GDPR or CCPA—into actionable technical requirements. This translation process was not straightforward. Many described lengthy interpretations, negotiations with legal teams, or trial-and-error

attempts to incorporate policies into system designs. This challenge was exacerbated by the shifting legal landscape: participants noted how new or evolving regulations required them to continuously adjust their interpretations and solutions, contributing to a sense of uncertainty.

Reporting structures. Interviewees highlighted how organizational positioning affected their ability to implement privacy measures. Some privacy engineers were embedded within security teams, while others reported to legal or product units. Participants observed that the department they were housed in influenced the kind of support they received, as well as their perceived authority. For instance, privacy engineers embedded in legal teams found it easier to leverage regulatory mandates, but struggled when proposing technical changes. Conversely, those in engineering departments found it simpler to influence system design but more difficult to secure buy-in from legal experts. Ultimately, these structural arrangements shaped their day-to-day interactions and determined how readily they could advocate for privacy initiatives.

Competencies and evaluation

Finally, participants also discussed a range of competencies they believed were essential for privacy engineers. They also reflected on how their performance was assessed, both formally and informally. Four key themes emerged: the common reliance on informal evaluation methods, the necessity of cross-functional stakeholder management, the importance of risk and threat management expertise, and the centrality of conducting privacy reviews and offering informed advice.

Informal evaluation. When asked how their performance was assessed, participants often described informal, ad hoc evaluation processes. Rather than a standardized set of metrics, many said that their work was gauged through conversations, internal feedback, or their ability to respond effectively to emerging privacy risks. Some participants noted that their success was measured by the absence of privacy incidents, an outcome that can be difficult to attribute directly to their interventions. Others mentioned that peer recognition, project successes, and positive stakeholder feedback were the main indicators that they were meeting organizational expectations.

Cross-functional stakeholder management. Participants emphasized the importance of working effectively with a wide array of stakeholders, including product teams, legal counsel, marketing departments, user researchers, and external clients. They detailed how successful privacy engineering demands not only technical acumen but also communication, negotiation, and diplomacy. These interactions required them to explain complex technical concepts to non-technical audiences, manage conflicting priorities, and, at times, negotiate compromises that balanced privacy with other organizational needs. Ensuring broad align-

ment and shared understanding was considered a critical competency that directly influenced their impact and the success of privacy initiatives.

Risk and threat management. Handling privacy threats and assessing potential risks emerged as central responsibilities. Participants described their work as identifying vulnerabilities, evaluating data handling practices, and recommending mitigations before issues escalated. Many used privacy threat modeling or "red teaming" exercises to anticipate potential problems. Their competence in risk management was closely tied to their ability to translate abstract threats into concrete technical recommendations. Some participants recounted how demonstrating foresight and providing actionable solutions contributed significantly to their credibility within the organization.

Perform reviews and offer advice. Many participants mentioned that a significant portion of their workload involved conducting privacy reviews, performing assessments of systems or products, and offering tailored advice. These activities often took the form of providing input on design documents, reviewing feature proposals, or helping product teams understand privacy implications. Participants reported that offering timely, accurate, and actionable guidance was one of their most visible contributions, serving as a tangible sign of their value as in-house privacy experts. They emphasized the importance of clarity, accessibility, and practicality in their advice, ensuring that stakeholders could implement recommended changes with minimal confusion.

7.5 Discussion

In this section, we reflect on our findings and situate them within the broader context of the evolving field of privacy engineering. We revisit the key themes identified in our analysis—conceptual ambiguity, motivational drivers, organizational and regulatory challenges, and essential competencies—and discuss their implications for practitioners, organizations, regulators, and researchers. We conclude by outlining practical recommendations, acknowledging limitations, and suggesting directions for future research.

Interpreting the Role of Privacy Engineering in an Evolving Landscape

Our study revealed that privacy engineering remains a nascent and fluid discipline, one whose boundaries and practices continue to crystallize. Participants consistently noted that their roles are not strictly defined, reflecting a broader industry reality in which privacy engineers must adapt to varied organizational structures, team compositions, and product domains. This ambiguity underscores a critical need for more formalized frameworks, job descriptions, and skillset benchmarks that allow practitioners, recruiters, and organizational leaders to

better understand what privacy engineers do, how they add value, and where they fit within existing hierarchies.

Compared to earlier conceptualizations of privacy engineering that focus on technical solutions or legal compliance alone, our participants' experiences highlight a more holistic view. Privacy engineering emerges not as a narrow specialization but rather as a multidisciplinary pursuit. Privacy engineers blend technical expertise with strong communication and cross-functional coordination capabilities. As privacy concerns increase in scope and complexity, the ability to align technical solutions with regulatory mandates, stakeholder values, and organizational priorities becomes a key differentiator. This multifaceted identity suggests that privacy engineering is both a technical craft and a socio-technical mediation role, one that will continue to gain prominence as data-driven business models evolve.

Motivations and the Ethical Dimension of Privacy Engineering

Our findings suggest that privacy engineers are often driven by more than professional ambition or technological interest. Many participants reported personal, ethical motivations for engaging in privacy work—seeing their role as safeguarding user rights and promoting responsible data stewardship. This moral dimension sets privacy engineers apart from more traditional engineering roles focused primarily on functionality, performance, or security. The engineers' emphasis on ethical considerations aligns with increasing public scrutiny and heightened expectations for responsible data use, trustworthiness, and fairness.

This ethical orientation can have several positive impacts. It may help organizations anticipate reputational risks and enhance brand trust by embedding privacy values early in the development process. At the same time, this dimension calls for careful organizational support. Privacy engineers who view their work as ethically significant may become disheartened if faced with persistent misaligned incentives. Organizations must therefore create an environment that acknowledges and leverages privacy engineers' moral commitments—through policies, reward structures, and leadership support—to ensure that these motivations translate into sustainable privacy outcomes.

Organizational and Regulatory Challenges

Participants highlighted persistent organizational barriers, including the difficulty in securing the support of product teams or executives primarily focused on revenue and rapid deployment. Misaligned incentives remain a core obstacle: while regulators and public sentiment increasingly demand proactive privacy measures, internal metrics often fail to recognize or reward preventive privacy work. Addressing these challenges may require organizations to adopt new performance indicators, reconfigure reporting lines, or invest in privacy education to ensure that privacy engineering is not perceived as a mere compliance cost.

Regulatory complexity emerged as another key challenge, with privacy engineers struggling to translate shifting, high-level legal requirements into concrete technical practices.

This issue points to a need for clearer guidance from policymakers, greater collaboration between legal and engineering teams, and the development of standardized industry frameworks to operationalize legal mandates. Over time, the field could benefit from consolidated guidelines that help privacy engineers bridge the gap between legal text and system design—potentially reducing the uncertainty and resource expenditures involved in continuous regulatory interpretation.

Competencies and Informal Evaluations

Our analysis showed that privacy engineers rely on a mix of technical skills, cross-functional communication abilities, and risk management competencies. While technical proficiency is a cornerstone, especially in identifying and integrating privacy-enhancing technologies, the softer yet equally crucial skill of stakeholder management emerged as vital. Privacy engineers must negotiate among multiple parties—developers, product owners, legal experts, and marketing teams—to achieve privacy goals. This finding strengthens the case for conceptualizing privacy engineering as a role that thrives at the intersection of technology and organizational culture.

However, participants also noted that their performance often goes unmeasured by formal metrics, relying instead on informal feedback, trust, and the absence of incidents. For organizations to foster more systematic improvement, clearer evaluation frameworks could be established. These might include measures tied to the integration of privacy controls in product roadmaps, user satisfaction with data handling practices, or the number and severity of privacy-related incidents avoided. More robust assessment methods could provide privacy engineers with recognition and career advancement paths, reinforcing their strategic value to the organization.

Implications for Industry and Practice

The insights from this study offer several practical recommendations. First, organizations may benefit from developing standardized role descriptions and career ladders for privacy engineers, clarifying expectations and needed skills. Doing so can facilitate recruitment and help new hires integrate more smoothly. Second, incorporating privacy-focused key performance indicators (KPIs) into product development cycles can align incentives, ensuring that proactive privacy measures are recognized and rewarded. Third, regular training sessions for both engineers and non-technical stakeholders can enhance organizational privacy literacy, improving communication and collaboration.

Another implication is the potential role of privacy engineers as internal advocates and educators. Organizations might formalize this function, encouraging privacy engineers to hold periodic workshops or office hours for developers, product managers, and designers. Such activities could help embed a culture of privacy by design and reinforce compliance as a shared responsibility rather than a niche concern.

Research Opportunities and Future Directions

Our study lays a foundation for further inquiry. While we identified common themes and challenges, future research could quantify their prevalence across different sectors, organizational sizes, and cultural contexts. Surveys and larger-scale quantitative studies could complement our qualitative findings, enabling more generalizable conclusions. Longitudinal studies might also observe how the role of privacy engineers evolves as regulations, technologies, and societal norms continue to shift.

Another avenue for research involves exploring the interplay between privacy engineering and emerging areas such as machine learning fairness, explainability, and data governance. As organizations increasingly rely on data-intensive processes, privacy engineers might collaborate with professionals addressing algorithmic transparency or responsible AI development. Studying these intersections could inform a more holistic approach to data ethics, trustworthiness, and accountability.

Finally, more nuanced regulatory and policy-oriented research could examine whether and how policymakers might rely on insights from privacy engineers to refine legal frameworks and provide clearer, more practical guidance. Such a feedback loop could help harmonize legal requirements with engineering realities, ultimately producing more effective privacy protections for users.

Limitations

Our study is qualitative and exploratory, and while our participant pool was larger than that recommended by some qualitative guidelines, it remains subject to self-selection biases. Those who chose to participate may be particularly motivated or established privacy engineers. Moreover, our focus was limited primarily to the United States context, and most participants were based in California—a major technology hub. Privacy engineering roles and perceptions may differ in other cultural or regulatory environments.

Additionally, the dynamic nature of privacy engineering means that our findings offer a snapshot in time. Roles, definitions, and practices may shift as the field matures, as regulatory regimes change, or as new technologies emerge. Future studies could revisit these themes periodically to track the evolution of privacy engineering.

7.6 Conclusion

This research provides a rare, in-depth look into the realities of privacy engineering roles, uncovering the complexities and nuances that shape this emerging field. Far from being a narrowly defined technical job, privacy engineering involves blending robust technical foundations with a flexible understanding of legal and ethical considerations, organizational cultures, and collaborative communication skills. Privacy engineers act as linchpins, translating

high-level privacy principles into concrete, system-level implementations and ensuring that user rights and trust are respected in an increasingly data-driven world.

As privacy considerations move to the forefront of regulatory and consumer attention, the importance of well-defined, well-resourced privacy engineering functions will only grow. By illuminating how privacy engineers understand their roles, what motivates them, and what challenges they face, this study aims to help organizations, policymakers, and educators recognize the critical value of privacy engineering. With greater clarity, structured support, and meaningful incentives, privacy engineers can more effectively guide the development of responsible, compliant, and user-centric digital products—ultimately shaping a more trustworthy and privacy-aware technological landscape.

Chapter 8

Conclusion

This dissertation set out to understand why developers consistently fail to embed privacy requirements into software and how professionalizing privacy engineering can help address these systemic issues. The research began by examining the evolving legal and regulatory landscapes that frame modern privacy obligations. It argued that while data protection principles and regulations like the GDPR and CCPA provide concrete guidance, they leave organizations and developers struggling with interpretation, fragmentation, and implementation challenges. The result is a persistent gap between what privacy laws mandate and what software systems deliver, with users bearing the cost of these shortcomings through privacy breaches and data misuse.

8.1 Key Contributions and Findings

The research in this thesis identified several key insights:

Complexity of Privacy Obligations. Modern privacy regulations, such as the GDPR and CCPA, impose a wide array of requirements—from data minimization and purpose specification to granular user rights like data access and deletion. These obligations are notoriously difficult to translate into engineering practices. The conceptual ambiguity of privacy, coupled with fragmented regulations across jurisdictions, contributes to confusion and non-compliance.

Systemic Developer Failures. Traditional software development processes often treat privacy as an afterthought. Chapters 4 and 5 illustrated how developers, when left without proper guidance and structures, produce code and use third-party services in ways that inadvertently violate privacy principles. Empirical evidence from these measurement studies showed widespread undisclosed data collection, incomplete responses to user rights requests, and personal information inadvertently sent to third-party providers. These examples high-

light how standard incentives, lack of specialized knowledge, and insufficient integration of privacy requirements lead to system-level failures.

Role of Professionalized Privacy Engineering. The dissertation makes a case for "professionalizing" privacy engineering. By this, it does not merely mean hiring technologists with some privacy knowledge. Instead, it points to creating well-defined roles for privacy engineers who have specialized skills to navigate the complex regulatory environment, translate abstract rules into actionable technical requirements, and embed data protection principles directly into software architecture and design.

Privacy engineers stand apart from security engineers or compliance officers by combining technical acumen with a nuanced understanding of privacy laws and ethical considerations. They serve as mediators between legal expectations and software implementations, bridging the communication gap between policy-oriented stakeholders and technical teams. Their involvement can transform privacy from an external requirement into a built-in feature of the software development lifecycle.

Organizational and Incentive Structures. The interviews with practicing privacy engineers (Chapter 7) underscored the necessity of supportive organizational structures. Even skilled privacy engineers struggle to operate effectively within environments that reward rapid feature releases and cost-cutting over long-term trust and regulatory compliance. Recognizing the strategic value of privacy and investing in privacy engineering roles—through executive buy-in, training, cross-functional collaboration, and measurable incentives—can ensure that privacy is not just a legal checkbox but a key component of product quality.

Measuring Success and Accountability. The dissertation identified that current metrics for success in privacy compliance are often too vague or backward-looking. By the time a privacy violation surfaces, the harm is done. Professionalizing privacy engineering entails developing proactive measures, from systematic privacy impact assessments to continuous verification and compliance checks integrated into the SDLC. Just as quality assurance and security testing have become standard practice, so too can ongoing privacy validation. Moreover, documenting privacy-by-design decisions and involving privacy engineers in design reviews ensures accountability and reduces the risk of unforeseen lapses.

8.2 Implications for Industry and Policy

The findings hold significant implications for businesses, developers, regulators, and the broader ecosystem of stakeholders concerned with user trust and digital ethics:

For Businesses and Developers. Organizations should invest in privacy engineers as integral members of product teams. Companies benefit when privacy engineering is not

relegated to siloed groups but fully integrated into the requirements analysis, architecture design, and testing phases. Aligning incentives, providing training, and recognizing the strategic value of privacy can foster a culture where engineers anticipate privacy needs rather than retrofitting fixes under legal duress.

For Regulators and Policymakers. For Regulators and Policymakers: While laws like GDPR and CCPA set robust privacy standards, they provide less guidance on how to operationalize these requirements technically. Policymakers can facilitate compliance by offering clear technical guidelines, frameworks, or accreditation for privacy engineers. Such steps would provide a baseline for industry practice, reduce confusion, and enable more consistent enforcement. Encouraging the formalization of privacy engineering certification, education, and professional standards would help both companies and regulators converge on best practices.

For the Broader Technology Ecosystem. The systemic issues identified in this dissertation reflect a broader challenge of embedding ethical considerations into the very fabric of technology development. Privacy engineering can serve as a model for other emergent fields—such as bias in AI or sustainability in computing—where clear professional roles, standards, and methodologies are similarly needed. By championing the professionalization of these roles, the tech ecosystem can better align product innovation with societal values.

8.3 Limitations and Future Work

This dissertation focused on privacy obligations primarily shaped by laws and norms from Western jurisdictions. Future research could broaden the scope to account for cultural variations in privacy expectations and diverging legal standards worldwide. Additionally, the measurement studies and interviews examined here represent particular market segments and organizational contexts. As privacy engineering roles proliferate, additional longitudinal and comparative studies will be valuable, enabling the community to refine definitions, measure effectiveness more concretely, and track the evolving responsibilities of privacy engineers.

The dissertation also identified gaps in tooling and standardized methodologies. Further research could develop and evaluate developer-centric tools, pattern libraries, or privacy design languages. Comparative studies between organizations with mature privacy engineering functions and those still reliant on ad hoc approaches could further substantiate the claimed benefits of professionalizing privacy engineering.

This dissertation highlights that achieving meaningful privacy protection in software systems is neither accidental nor trivial. It requires deliberate integration of privacy principles from the inception of product design, supported by professionals trained to navigate regulatory complexities and technical constraints. The professionalization of privacy engineering emerges as both a pragmatic solution and a guiding framework for the future. When pri-

vacy engineering is recognized as a distinct and essential profession—analogous to security or reliability engineering—organizations gain a structured approach to preventing privacy failures, regulators gain more coherent compliance pathways, and, most importantly, users gain software products that respect and preserve their fundamental rights.

In a rapidly evolving digital landscape, the professionalization of privacy engineering is not just a remedy for current systemic failures, but a forward-looking strategy that lays the groundwork for trust, accountability, and respect for user autonomy in the technologies of tomorrow.

Bibliography

[1] European Data Protection Board (EDPB). *1.2 billion euro fine for Facebook as a result of EDPB binding decision*. https://www.edpb.europa.eu/news/news/2023/12-billion-euro-fine-facebook-result-edpb-binding-decision_en. 2023.

[2] Yasemin Acar, Michael Backes, Sascha Fahl, Simson Garfinkel, Doowon Kim, Michelle L Mazurek, and Christian Stransky. "Comparing the usability of cryptographic apis". In: *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2017, pp. 154–171.

[3] Alessandro Acquisti, Curtis Taylor, and Liad Wagman. "The economics of privacy". In: *Journal of economic Literature* 54.2 (2016), pp. 442–492.

[4] Supriya Adhatarao, Cédric Lauradoux, and Cristiana Santos. "Why IP-based Subject Access Requests Are Denied?" In: *arXiv preprint arXiv.2103.01019* (2021).

[5] Mansour Ahmadi, Battista Biggio, Steven Arzt, Davide Ariu, and Giorgio Giacinto. "Detecting misuse of google cloud messaging in android badware". In: *Proceedings of the 6th Workshop on Security and Privacy in Smartphones and Mobile Devices*. 2016, pp. 103–112.

[6] AirShip. *Android SDK Setup*. https://docs.airship.com/platform/mobile/setup/sdk/android/. (Accessed on 10/10/2023). 2023.

[7] Fatemeh Alizadeh, Timo Jakobi, Alexander Boden, Gunnar Stevens, and Jens Boldt. "GDPR reality check-claiming and investigating personally identifiable data from companies". In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. New York, NY, USA: IEEE, 2020, pp. 120–129.

[8] Gonçalo Almeida Teixeira, Miguel Mira da Silva, and Ruben Pereira. "The critical success factors of GDPR implementation: a systematic literature review". In: *Digital Policy, Regulation and Governance* 21.4 (2019), pp. 402–418.

[9] Noura Alomar and Serge Egelman. "Developers say the darnedest things: Privacy compliance processes followed by developers of child-directed apps". In: *Proceedings on Privacy Enhancing Technologies* 4.2022 (2022), p. 24.

[10] R. Anderson. "Why information security is hard - an economic perspective". In: *Seventeenth Annual Computer Security Applications Conference*. 2001, pp. 358–365. DOI: 10.1109/ACSAC.2001.991552.

[11] Benjamin Andow, Samin Yaseer Mahmud, Wenyu Wang, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Tao Xie. "PolicyLint: Investigating Internal Privacy Policy Contradictions on Google Play". In: *28th USENIX security symposium (USENIX security 19)*. Berkeley, CA, USA: USENIX, 2019, pp. 585–602.

[12] Benjamin Andow, Samin Yaseer Mahmud, Justin Whitaker, William Enck, Bradley Reaves, Kapil Singh, and Serge Egelman. "Actions Speak Louder than Words:Entity-Sensitive Privacy Policy and Data Flow Analysis with PoliCheck". In: *29th USENIX Security Symposium (USENIX Security 20)*. Berkeley, CA, USA: USENIX, 2020, pp. 985–1002.

[13] Apple. *Notifications Overview*. Apple Developer. https://developer.apple.com/notifications/. 2023.

[14] Apple. *Push Token Requests*. https://www.apple.com/legal/transparency/push-token.html. (Accessed on 06/01/2024). 2023.

[15] Apple Inc. *Generating a remote notification*. https://developer.apple.com/documentation/usernotifications/setting_up_a_remote_notification_server/generating_a_remote_notification. (Accessed on 10/10/2023). 2023.

[16] Internet Archive. *Wayback Machine*. https://archive.org/. (Accessed on 10/10/2023). 2023.

[17] Renana Arizon-Peretz, Irit Hadar, Gil Luria, and Sofia Sherman. "Understanding developers' privacy and security mindsets via climate theory". In: *Empirical Software Engineering* 26 (2021), pp. 1–43.

[18] Jef Ausloos and Pierre Dewitte. "Shattering One-Way Mirrors. Data Subject Access Rights in Practice". In: *Data Subject Access Rights in Practice (January 20, 2018). International Data Privacy Law* 8.1 (2018), pp. 4–28.

[19] Welfare. Secretary's Advisory Committee on Automated Personal Data Systems. *Records, Computers, and the Rights of Citizens: Report*. Vol. 10. US Department of Health, Education & Welfare, 1973.

[20] John Babikian. "Securing Rights: Legal Frameworks for Privacy and Data Protection in the Digital Era". In: *Law Research Journal* 1.2 (2023), pp. 91–101.

[21] Rebecca Balebako, Abigail Marsh, Jasmine Lin, Jason Hong, and Lorrie F. Cranor. "The privacy and security behaviors of smartphone app developers". In: *Workshop on Usable Security*. Reston, VA, USA: The Internet Society, 2014, pp. 1–10.

[22] Kayce Basques and Matt Gaunt. *Push notifications overview*. https://web.dev/articles/push-notifications-overview. (Accessed on 10/10/2023). 2023.

[23] BCC Research. *Digital Advertising: Global Market Opportunities and Forecast to 2027*. https://www.bccresearch.com/market-research/information-technology/digital-advertising-market.html. Apr. 2023.

[24] Kathrin Bednar, Sarah Spiekermann, and Marc Langheinrich. "Engineering Privacy by Design: Are engineers ready to live up to the challenge?" In: *The Information Society* 35.3 (2019), pp. 122–142.

[25] Colin J Bennett and Charles D Raab. *The governance of privacy: Policy instruments in global perspective*. Routledge, 2017.

[26] Michael Birnhack, Eran Toch, and Irit Hadar. "Privacy mindset, technological mindset". In: *Jurimetrics* 55 (2014), p. 55.

[27] Android Developers Blog. *Project Capillary: End-to-end encryption for push messaging, simplified*. https://android-developers.googleblog.com/2018/06/project-capillary-end-to-end-encryption.html. (Accessed on 10/10/2023). June 2018.

[28] Coline Boniface, Imane Fouad, Nataliia Bielova, Cédric Lauradoux, and Cristiana Santos. "Security analysis of subject access request procedures". In: *Annual Privacy Forum*. Berlin, Germany: Springer, 2019, pp. 182–209.

[29] Alex Bowyer, Jack Holt, Josephine Go Jefferies, Rob Wilson, David Kirk, and Jan David Smeddinck. "Human-GDPR Interaction: Practical Experiences of Accessing Personal Data". In: *CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2022, pp. 1–19.

[30] Sean Brooks, Sean Brooks, Michael Garcia, Naomi Lefkovitz, Suzanne Lightman, and Ellen Nadeau. *An introduction to privacy engineering and risk management in federal systems*. 2017.

[31] Luca Bufalieri, Massimo La Morgia, Alessandro Mei, and Julinda Stefa. "GDPR: when the right to access personal data becomes a threat". In: *2020 IEEE International Conference on Web Services (ICWS)*. New York, NY, USA: IEEE, 2020, pp. 75–83.

[32] Duc Bui, Kang G Shin, Jong-Min Choi, and Junbum Shin. "Automated Extraction and Presentation of Data Practices in Privacy Policies." In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2021.2 (2021), pp. 88–110.

[33] Matt Burgess. *What is GDPR? The summary guide to GDPR compliance in the UK*. https://www.wired.co.uk/article/what-is-gdpr-uk-eu-legislation-compliance-summary-fines-2018. Wired. Mar. 2020.

[34] Matteo Cagnazzo, Thorsten Holz, and Norbert Pohlmann. "GDPiRated–stealing personal information on-and offline". In: *European Symposium on Research in Computer Security*. Berlin, Germany: Springer, 2019, pp. 367–386.

[35] California Consumer Privacy Act (CCPA). California Civil Code §1798.100 et seq. 2018.

[36] L. Cavallaro, P. Saxena, and R. Sekar. "On the Limits of Information Flow Techniques for Malware Analysis and Containment". In: *Proc. of DIMVA*. Springer-Verlag, 2008, pp. 143–163. URL: http://dx.doi.org/10.1007/978-3-540-70542-0_8.

[37] Ann Cavoukian. "Operationalizing privacy by design: a guide to implementing". In: *Commun ACM* 55.9 (2012), p. 7.

[38] Ann Cavoukian. "Privacy by design". In: (2009).

[39] Ann Cavoukian. *Privacy by Design: The 7 Foundational Principles*. Information and Privacy Commissioner of Ontario. Accessed: 2024-12-12. 2011. URL: https://privacy.ucsc.edu/resources/privacy-by-design---foundational-principles.pdf.

[40] Ann Cavoukian. "Privacy by design: the definitive workshop. A foreword by Ann Cavoukian, Ph. D". In: *Identity in the Information Society* 3.2 (2010), pp. 247–251.

[41] Grazia Cecere, Fabrice Le Guel, and Vincent Lefrere. "Economics of free mobile applications: personal data as a monetization strategy". In: *HAL, Post-Print, Sep* (2018).

[42] Kathy Charmaz. *Constructing grounded theory: A practical guide through qualitative analysis*. sage, 2006.

[43] Yangyi Chen, Tongxin Li, XiaoFeng Wang, Kai Chen, and Xinhui Han. "Perplexed messengers from the cloud: Automated security analysis of push-messaging integrations". In: *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. 2015, pp. 1260–1272.

[44] Michael Chui, Mena Issler, Roger Roberts, and Lareina Yee. *Technology trends outlook 2023*. http://dln.jaipuria.ac.in:8080/jspui/bitstream/123456789/14260/1/Mckinsey-technology-trends-outlook-2023.pdf. 2023.

[45] U.S. Federal Trade Commission. *Avast, Ltd.* https://www.ftc.gov/system/files/ftc_gov/pdf/Complaint-Avast.pdf. Feb. 2024.

[46] U.S. Federal Trade Commission. *Flo Health, Inc.* https://www.ftc.gov/legal-library/browse/cases-proceedings/192-3133-flo-health-inc. June 2021.

[47] United States Congress. *Children's Online Privacy Protection Act of 1998*. Pub. L. No. 105-277, 112 Stat. 2681–728, codified at 15 U.S.C. §§ 6501–6506. Enacted October 21, 1998. 1998. URL: https://www.govinfo.gov/content/pkg/USCODE-2019-title15/html/USCODE-2019-title15-chap91.htm.

[48] United States Congress. *Federal Trade Commission Act of 1914*. Pub. L. No. 63-203, 38 Stat. 717, codified at 15 U.S.C. §§ 41–58. Enacted September 26, 1914. 1914. URL: https://www.govinfo.gov/content/pkg/USCODE-2019-title15/pdf/USCODE-2019-title15-chap2-subchapI.pdf.

[49] United States Congress. *Gramm-Leach-Bliley Act*. Pub. L. No. 106-102, 113 Stat. 1338, codified at 15 U.S.C. §§ 6801 et seq. Enacted November 12, 1999. 1999. URL: https://www.govinfo.gov/content/pkg/PLAW-106publ102/pdf/PLAW-106publ102.pdf.

[50] United States Congress. *Health Insurance Portability and Accountability Act of 1996*. Pub. L. No. 104-191, 110 Stat. 1936. Enacted August 21, 1996. 1996. URL: https://www.govinfo.gov/content/pkg/PLAW-104publ191/pdf/PLAW-104publ191.pdf.

[51] United States Congress. *Privacy Act of 1974*. Public Law 93-579, 88 Stat. 1896. Codified at 5 U.S.C. § 552a. 1974. URL: https://www.govinfo.gov/content/pkg/STATUTE-88/pdf/STATUTE-88-Pg1896.pdf.

[52] Andrew Cormack. "Is the Subject Access Right Now Too Great a Threat to Privacy". In: *Eur. Data Prot. L. Rev.* 2 (2016), p. 15.

[53] California Superior Court. *People of the State of California v. Sephora USA, Inc.*, Case No. CGC-22-601380. 2022.

[54] Cox, Joseph. *Here's a Warrant Showing the U.S. Government is Monitoring Push Notifications*. https://www.404media.co/us-government-warrant-monitoring-push-notifications-apple-google-yahoo/. (Accessed on 06/01/2024). 2023.

[55] European Union Agency for Cybersecurity (ENISA). *Engineering Personal Data Sharing*. https://www.enisa.europa.eu/publications/engineering-personal-data-sharing. (Accessed on 06/01/2024). 2023.

[56] Cybersecurity and Infrastructure Security Agency (CISA). *Shifting the Balance of Cybersecurity Risk: Principles and Approaches for Secure by Design Software*. https://www.cisa.gov/sites/default/files/2023-10/SecureByDesign_1025_508c.pdf. (Accessed on 06/01/2024). 2023.

[57] 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". In: *European Union Agency for Network and Information Security (ENISA) report* (2015).

[58] DataGuidance and Future of Privacy FOrum. *Comparing Privacy Laws: GDPR v. CCPA*. https://fpf.org/wp-content/uploads/2018/11/GDPR_CCPA_Comparison-Guide.pdf. 2018.

[59] Mariano Di Martino, Isaac Meers, Peter Quax, Ken Andries, and Wim Lamotte. "Revisiting identification issues in GDPR 'Right Of Access' policies: a technical and longitudinal analysis". In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2022.2 (2022), pp. 95–113.

[60] Mariano Di Martino, Pieter Robyns, Winnie Weyts, Peter Quax, Wim Lamotte, and Ken Andries. "Personal Information Leakage by Abusing the GDPR'Right of Access'". In: *Fifteenth Symposium on Usable Privacy and Security (SOUPS 2019)*. Berkeley, CA, USA: USENIX, 2019, pp. 371–385.

[61] Organisation for Economic Co-operation and Development (OECD). *Recommendation of the Council concerning Guidelines Governing the Protection of Privacy and Transborder Flows of Personal Data*. OECD Publishing. Original 1980 Guidelines amended on 11 July 2013. 2013. URL: https://legalinstruments.oecd.org/en/instruments/OECD-LEGAL-0188.

[62] Organization for Economic Co-operation and Development. *OECD guidelines on the protection of privacy and transborder flows of personal data*. 1980.

[63] Ntional Bureau of Economic Research. *Why Are There So Few Public Companies in the U.S.?* https://www.nber.org/digest/sep15/why-are-there-so-few-public-companies-us. Accessed: 2022-08-31. 2021.

[64] Anirudh Ekambaranathan, Jun Zhao, and Max Van Kleek. ""Money makes the world go around": Identifying Barriers to Better Privacy in Children's Apps From Developers' Perspectives". In: *Proceedings of the 2021 CHI conference on human factors in computing systems*. 2021, pp. 1–15.

[65] Samsung Electronics. *Samsung Push Service*. https://play.google.com/store/apps/details?id=com.sec.spp.push. (Accessed on 06/01/2024). 2023.

[66] W. Enck, P. Gilbert, B. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth. "TaintDroid: An Information-flow Tracking System for Realtime Privacy Monitoring on Smartphones". In: *Proc. of the 9th USENIX conference on Operating systems design and implementation (OSDI)*. 2010, pp. 393–407.

[67] European Parliament and the Council of the European Union. *Directive 95/46/EC of the European Parliament and of the Council*. Official Journal of the European Communities, L 281. Repealed by Regulation (EU) 2016/679 (GDPR) as of 25 May 2018. 1995. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A31995L0046.

[68] European Parliament and the Council of the European Union. *Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation)*. Official Journal of the European Union, L 119. Entered into force on 24 May 2016 and applicable from 25 May 2018. 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[69] Ming Fan, Le Yu, Sen Chen, Hao Zhou, Xiapu Luo, Shuyue Li, Yang Liu, Jun Liu, and Ting Liu. "An empirical evaluation of GDPR compliance violations in Android mHealth apps". In: *2020 IEEE 31st international symposium on software reliability engineering (ISSRE)*. New York, NY, USA: IEEE, 2020, pp. 253–264.

[70] Álvaro Feal, Paolo Calciati, Narseo Vallina-Rodriguez, Carmela Troncoso, Alessandra Gorla, et al. "Angel or devil? a privacy study of mobile parental control apps". In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2020.2 (2020), pp. 314–335.

[71] Federal Trade Commision (FTC). *FTC Requires Zoom to Enhance its Security Practices as Part of Settlement*. https://www.ftc.gov/news-events/news/press-releases/2020/11/ftc-requires-zoom-enhance-its-security-practices-part-settlement. (Accessed on 01/01/2024). Nov. 2020.

[72] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner. "Android permissions: user attention, comprehension, and behavior". In: *Proceedings of the 8th Symposium on Usable Privacy and Security*. SOUPS '12. Washington, D.C.: ACM, 2012. ISBN: 978-1-4503-1532-6. DOI: 10.1145/2335356.2335360. URL: http://doi.acm.org/10.1145/2335356.2335360.

[73] Caitlin Fennessy. *Privacy engineering: The what, why and how*. https://iapp.org/news/a/privacy-engineering-the-what-why-and-how. 2019.

[74] Frederick P. Brooks, Jr. *The Mythical Man-Month: Essays on Software Engineering*. Addison-Wesley, 1975.

[75] Frida. https://frida.re/. 2022.

[76] Ruth Gavison. "Privacy and the Limits of Law". In: *The Yale law journal* 89.3 (1980), pp. 421–471.

[77] Robert Gellman. "Fair information practices: A basic history - Version 2.30". In: *Available at SSRN* (2024).

[78] C. Gibler, J. Crussell, J. Erickson, and H. Chen. "AndroidLeaks: Automatically Detecting Potential Privacy Leaks in Android Applications on a Large Scale". In: *Proc. of the 5th international conference on Trust and Trustworthy Computing (TRUST)*. Springer-Verlag, 2012, pp. 291–307.

[79] GizChina. *HARMONYOS IS NOW FIRMLY THE THIRD LARGEST MOBILE PHONE OPERATING SYSTEM*. https://www.gizchina.com/2023/05/20/harmonyos-is-now-firmly-the-third-largest-mobile-phone-operating-system/. (Accessed on 01/01/2024). Dec. 2023.

[80] Google. *BaseBundle*. Android Developers. https://developer.android.com/reference/android/os/BaseBundle. 2023.

[81] Google. *Design for Safety*. Google Developers. https://developer.android.com/quality/privacy-and-security. 2023.

[82] Google. *FirebaseMessagingService*. https://firebase.google.com/docs/reference/android/com/google/firebase/messaging/FirebaseMessagingService. (Accessed on 06/01/2024). 2023.

[83] Google. *Play Console Help: Provide information for Google Play's Data safety section*. https://support.google.com/googleplay/android-developer/answer/10787469. (Accessed on 06/01/2024). 2023.

[84] Google for Developers. *About FCM messages*. Developer documentation for Firebase. https://firebase.google.com/docs/cloud-messaging/concept-options. 2024.

[85] M. I. Gordon, D. Kim, J. Perkins, Gilhamy, N. Nguyenz, and M. Rinard. "Information-Flow Analysis of Android Applications in DroidSafe". In: *Proc. of NDSS Symposium*. 2015.

[86] Graham Greenleaf. "Global data privacy laws 2019: 132 national laws & many bills". In: *157 Privacy Laws & Business International Report* (2019).

[87] Greg Guest, Arwen Bunce, and Laura Johnson. "How many interviews are enough? An experiment with data saturation and variability". In: *Field methods* 18.1 (2006), pp. 59–82.

[88] Seda Gürses and Jose M Del Alamo. "Privacy engineering: Shaping an emerging field of research and practice". In: *IEEE Security & Privacy* 14.2 (2016), pp. 40–46.

[89] Seda Gürses, Carmela Troncoso, and Claudia Diaz. "Engineering privacy by design". In: *Computers, Privacy & Data Protection* 14.3 (2011), p. 25.

[90] Seda Gürses, Carmela Troncoso, and Claudia Diaz. "Engineering privacy by design reloaded". In: *Amsterdam Privacy Conference*. Vol. 21. 2015.

[91] Marco Gutfleisch, Jan H Klemmer, Niklas Busch, Yasemin Acar, M Angela Sasse, and Sascha Fahl. "How does usable security (not) end up in software products? results from a qualitative interview study". In: *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE. 2022, pp. 893–910.

[92] Hana Habib, Sarah Pearman, Jiamin Wang, Yixin Zou, Alessandro Acquisti, Lorrie Faith Cranor, Norman Sadeh, and Florian Schaub. "" It's a scavenger hunt": Usability of Websites' Opt-Out and Data Deletion Choices". In: *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2020, pp. 1–12.

[93] Irit Hadar, Tomer Hasson, Oshrat Ayalon, Eran Toch, Michael Birnhack, Sofia Sherman, and Arod Balissa. "Privacy by designers: software developers' privacy mindset". In: *Empirical Software Engineering* 23 (2018), pp. 259–289.

[94] Catherine Han, Irwin Reyes, Álvaro Feal, Joel Reardon, Primal Wijesekera, Narseo Vallina-Rodriguez, Amit Elazar, Kenneth A Bamberger, and Serge Egelman. "The price is (not) right: Comparing privacy in free and paid apps". In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2020.3 (2020), pp. 222–242.

[95] Marit Hansen, Meiko Jensen, and Martin Rost. "Protection goals for privacy engineering". In: *2015 IEEE Security and Privacy Workshops*. IEEE. 2015, pp. 159–166.

[96] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. "Polisis: Automated analysis and presentation of privacy policies using deep learning". In: *27th USENIX Security Symposium (USENIX Security 18)*. Berkeley, CA, USA: USENIX, 2018, pp. 531–548.

[97] Woodrow Hartzog. "What is privacy? That's the wrong question". In: *U. Chi. L. Rev.* 88 (2021), p. 1677.

[98] Harwell, Drew and Schaffer, Aaron. *The FBI's new tactic: Catching suspects with push alerts*. https://www.washingtonpost.com/technology/2024/02/29/push-notification-surveillance-fbi/. (Accessed on 06/01/2024). 2024.

[99] Dominik Herrmann and Jens Lindemann. "Obtaining personal data and asking for erasure: Do app vendors and website owners honour your privacy rights?" In: *arXiv preprint arXiv:1602.01804* (2016), pp. 1–15.

[100] Office of Human Research Protections. *What is Human Subjects Research?* https://www.hhs.gov/ohrp/sites/default/files/OHRP-HHS-Learning-Module-Lesson2.pdf. 2022.

[101] Sangwon Hyun, Junsung Cho, Geumhwan Cho, and Hyoungshick Kim. "Design and analysis of push notification-based malware on android". In: *Security and Communication Networks* 2018 (2018).

[102] Jim Isaak and Mina J Hanna. "User data privacy: Facebook, Cambridge Analytica, and privacy protection". In: *Computer* 51.8 (2018), pp. 56–59.

[103] Priyank Jain, Manasi Gyanchandani, and Nilay Khare. "Big data privacy: a technological perspective and review". In: *Journal of Big Data* 3 (2016), pp. 1–25.

[104] Shubham Jain, Janne Lindqvist, et al. "Should I protect you? Understanding developers' behavior to privacy-preserving APIs". In: *Workshop on Usable Security (USEC'14)*. 2014.

[105] Laura Jehl and Alan Friel. *CCPA and GDPR Comparison Chart*. https://www.bakerlaw.com/webfiles/Privacy/2018/Articles/CCPA-GDPR-Chart.pdf. 2018.

[106] Qiwei Jia, Lu Zhou, Huaxin Li, Ruoxu Yang, Suguo Du, and Haojin Zhu. "Who leaks my privacy: Towards automatic and association detection with gdpr compliance". In: *International Conference on Wireless Algorithms, Systems, and Applications*. Berlin, Germany: Springer, 2019, pp. 137–148.

[107] joke2k. *Faker*. https://pypi.org/project/Faker/. Accessed: March 15, 2023. Mar. 2023.

[108] Scott Jordan, Yoshimichi Nakatsuka, Ercan Ozturk, Andrew Paverd, and Gene Tsudik. "Viceroy: Gdpr-/ccpa-compliant enforcement of verifiable accountless consumer requests". In: *arXiv preprint arXiv:2105.06942* (2021), pp. 1–17.

[109] JusTalk. *Is it safe to use JusTalk?* https://web.archive.org/web/20230407183707/https://justalk.com/support/general/g6. (Accessed on 10/10/2023). 2023.

[110] P. G. Kelley, L. F. Cranor, and N. Sadeh. "Privacy as part of the app decision-making process". In: *Proceedings of the SIGCHI conference on human factors in computing systems*. 2013, pp. 3393–3402.

[111] Katharine Kemp. "Concealed data practices and competition law: why privacy matters". In: *European Competition Journal* 16.2-3 (2020), pp. 628–672.

[112] Gabriel Khoury. *A Mass Exodus from WhatsApp to Signal and Other Privacy-focused Messaging Apps May have been Misinformed*. https://georgetownlawtechreview.org/a-mass-exodus-from-whatsapp-to-signal-and-other-privacy-focused-messaging-apps-may-have-been-misinformed/GLTR-02-2021/. 2021.

[113] J. Kim, Y. Yoon, K. Yi, and J. Shin. "ScanDal: Static Analyzer for Detecting Privacy Leaks in Android Applications". In: *IEEE Workshop on Mobile Security Technologies (MoST)* (2012).

[114] Simon Koch, Malte Wessels, Benjamin Altpeter, Madita Olvermann, and Martin Johns. "Keeping privacy labels honest". In: *Proceedings on Privacy Enhancing Technologies* 4.486-506 (2022), pp. 2–2.

[115] Konev, Max. *Statement on the Reuters Story Regarding Pushwoosh*. https://blog.pushwoosh.com/blog/statement-on-the-reuters-story-regarding-pushwoosh/. (Accessed on 06/01/2024). 2022.

[116] Bert-Jaap Koops, Bryce Clayton Newell, Tjerk Timan, Ivan Skorvanek, Tomislav Chokrevski, and Masa Galic. "A typology of privacy". In: *U. Pa. J. Int'l L.* 38 (2016), p. 483.

[117] Blagovesta Kostova, Seda Gürses, and Carmela Troncoso. "Privacy engineering meets software engineering. on the challenges of engineering privacy bydesign". In: *arXiv preprint arXiv:2007.08613* (2020).

[118] Klaus Krippendorff. *Computing Krippendorff's Alpha-Reliability*. 2011. URL: https://repository.upenn.edu/asc_papers/43.

[119] Klaus Krippendorff. *Content analysis: An introduction to its methodology*. Thousand Oaks, CA, USA: Sage publications, 2018.

[120] Jacob Leon Kröger, Jens Lindemann, and Dominik Herrmann. "How do app vendors respond to subject access requests? A longitudinal privacy study on iOS and Android Apps". In: *Proceedings of the 15th International Conference on Availability, Reliability and Security*. New York, NY, USA: ACM, 2020, pp. 1–10.

[121] Sophie Kuebler-Wachendorff, Robert Luzsa, Johann Kranz, Stefan Mager, Emmanuel Syrmoudis, Susanne Mayr, and Jens Grossklags. "The Right to Data Portability: conception, status quo, and future directions". In: *Informatik Spektrum* 44.4 (2021), pp. 264–272.

[122] Hayoung Lee, Taeho Kang, Sangho Lee, Jong Kim, and Yoonho Kim. "Punobot: Mobile botnet using push notification service in android". In: *Information Security Applications: 14th International Workshop, WISA 2013, Jeju Island, Korea, August 19-21, 2013, Revised Selected Papers 14*. Springer. 2014, pp. 124–137.

[123] California State Legislature. *California Consumer Privacy Act of 2018 (CCPA)*. California Civil Code §1798.100–1798.199. Enacted by Stats. 2018, Ch. 55, Sec. 3 (AB 375). 2018. URL: https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5.

[124] 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". In: *Proceedings of the ACM on Human-Computer Interaction* 4.CSCW3 (2021), pp. 1–28.

[125] Tongxin Li, Xiaoyong Zhou, Luyi Xing, Yeonjoon Lee, Muhammad Naveed, XiaoFeng Wang, and Xinhui Han. "Mayhem in the push clouds: Understanding and mitigating security hazards in mobile push-messaging services". In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. 2014, pp. 978–989.

[126] Thomas Linden, Rishabh Khandelwal, Hamza Harkous, and Kassem Fawaz. "The privacy policy landscape after the GDPR". In: *arXiv preprint arXiv:1809.08396* (2018), pp. 1–18.

[127] Tianming Liu, Haoyu Wang, Li Li, Guangdong Bai, Yao Guo, and Guoai Xu. "Dapanda: Detecting aggressive push notifications in android apps". In: *2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)*. IEEE. 2019, pp. 66–78.

[128] Pierpaolo Loreti, Lorenzo Bracciale, and Alberto Caponi. "Push attack: binding virtual and real identities using mobile push notifications". In: *Future Internet* 10.2 (2018), p. 13.

[129] Jiadong Lou, Xiaohan Zhang, Yihe Zhang, Xinghua Li, Xu Yuan, and Ning Zhang. "Devils in Your Apps: Vulnerabilities and User Privacy Exposure in Mobile Notification Systems". In: *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*. IEEE. 2023, pp. 28–41.

[130] Linda A Macaulay. *Requirements engineering*. Springer Science & Business Media, 2012.

[131] Mary Madden. *Public Perceptions of Privacy and Security in the Post-Snowden Era*. Pew Research Center. https://www.pewresearch.org/internet/2014/11/12/public-privacy-perceptions/. Nov. 2014.

[132] Maureen Mahoney. *California Consumer Privacy Act: Are consumers' digital rights protected*. Tech. rep. Consumer Reports Digital Lab, 2020.

[133] Karl Marx. *Critique of the Gotha program*. 1875.

[134] Jonathan Mayer. *Princeton-Radboud Study on Privacy Law Implementation*. https://privacystudy.cs.princeton.edu/. Dec. 2021.

[135] Jim McCambridge, John Witton, and Diana R Elbourne. "Systematic review of the Hawthorne effect: new concepts are needed to study research participation effects". In: *Journal of clinical epidemiology* 67.3 (2014), pp. 267–277.

[136] Colleen McClain, Michelle Faverio, Monica Anderson, and Eugenie Park. "How Americans view data privacy". In: *Pew Research Center* (2023).

[137] Microsoft. *What are Skype Private Conversations?* https://web.archive.org/web/20230606085952/https://support.skype.com/en/faq/fa34824/what-are-skype-private-conversations. (Accessed on 10/10/2023). 2023.

[138] Müge Fazlioglu. *IAPP Privacy and Consumer Trust Report*. https://iapp.org/resources/article/privacy-and-consumer-trust-summary/. (Accessed on 12/12/2024). 2023.

[139] Deirdre K Mulligan, Colin Koopman, and Nick Doty. "Privacy is an essentially contested concept: a multi-dimensional analytic for mapping privacy". In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 374.2083 (2016), p. 20160118.

[140] National Conference of State Legislatures (NCSL). *Consumer Data Privacy Legislation*. Accessed: 2024-12-12. 2023. URL: https://www.ncsl.org/technology-and-communication/consumer-data-privacy.

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

[142] Helen Nissenbaum. "Privacy as contextual integrity". In: *Wash. L. Rev.* 79 (2004), p. 119.

[143] Helen Nissenbaum. *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. 2009.

[144] Clive Norris and Xavier L'Hoiry. "Exercising Citizen Rights Under Surveillance Regimes in Europe–Meta-analysis of a Ten Country Study". In: *The Unaccountable State of Surveillance*. Berlin, Germany: Springer, 2017, pp. 405–455.

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

[146] OneSignal. *Firebase Cloud Messaging (FCM) Compared to OneSignal*. https://web.archive.org/web/20230603040346/https://onesignal.com/blog/firebase-vs-onesignal/. (Accessed on 10/10/2023). 2023.

[147] OneSignal. *What is a push notifications service and how does it work?* https://onesignal.com/blog/what-is-a-push-notifications-service-and-how-does-it-work/. (Accessed on 2/23/24). 2023.

[148] OWASP. *Zed Attack Proxy (ZAP)*. https://www.zaproxy.org/. 2022.

[149] Frank Pallas, Katharina Koerner, Isabel Barberá, Jaap-Henk Hoepman, Meiko Jensen, Nandita Rao Narla, Nikita Samarin, Max-R Ulbricht, Isabel Wagner, Kim Wuyts, et al. "Privacy Engineering From Principles to Practice: A Roadmap". In: *IEEE Security & Privacy* 22.2 (2024), pp. 86–92.

[150] Charlie Parker, Sam Scott, and Alistair Geddes. "Snowball sampling". In: *SAGE research methods foundations* (2019).

[151] James Pavur and Casey Knerr. "Gdparrrrr: Using privacy laws to steal identities". In: *arXiv preprint arXiv:1912.00731* (2019), pp. 1–10.

[152] James Pearson and Marisa Taylor. *Russian software disguised as American finds its way into U.S. Army, CDC apps*. https://www.reuters.com/technology/exclusive-russian-software-disguised-american-finds-its-way-into-us-army-cdc-2022-11-14/. (Accessed on 06/01/2024). 2022.

[153] Mariana Peixoto, Dayse Ferreira, Mateus Cavalcanti, Carla Silva, Jéssyka Vilela, João Araújo, and Tony Gorschek. "On understanding how developers perceive and interpret privacy requirements research preview". In: *Requirements Engineering: Foundation for Software Quality: 26th International Working Conference, REFSQ 2020, Pisa, Italy, March 24–27, 2020, Proceedings 26*. Springer. 2020, pp. 116–123.

[154] Benjamin William Perry and Rachel M. LaBruyere. *Who Has My Data? EU Court Rules GDPR Requires Disclosure of Data Recipient Identities, Not Just Categories, in Response to Data Subject Access Requests*. Lexology. https://www.lexology.com/library/detail.aspx?g=5d58c7be-ca44-4249-b11b-1a68144dac24. Feb. 2023.

[155] Google Play. *WeChat: About this app*. https://web.archive.org/web/20230323082225/https://play.google.com/store/apps/details?id=com.tencent.mm&hl=en_US&gl=US. (Accessed on 10/10/2023). 2023.

[156] Roger S Pressman. "Software Engineering: a practitioner's approach". In: *Pressman and Associates* (2005).

[157] Maxwell Prybylo, Sara Haghighi, Sai Teja Peddinti, and Sepideh Ghanavati. "Evaluating Privacy Perceptions, Experience, and Behavior of Software Development Teams". In: *arXiv preprint arXiv:2404.01283* (2024).

[158] Pusher. *Configure FCM*. https://pusher.com/docs/beams/getting-started/android/configure-fcm/. (Accessed on 10/10/2023). 2023.

[159] Random Lists. *Random Lists*. https://www.randomlists.com/. Accessed: March 15, 2023. 2023.

[160] A. Razaghpanah, A. A. Niaki, N. Vallina-Rodriguez, S. Sundaresan, J. Amann, and P. Gill. "Studying TLS usage in Android apps". In: *Proceedings of the 13th International Conference on emerging Networking EXperiments and Technologies*. 2017, pp. 350–362.

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

[162] 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)*. Reston, VA, USA: Internet Society, 2018.

[163] Irwin Reyes, Primal Wijesekera, Joel Reardon, Amit Elazari Bar On, Abbas Razaghpanah, Narseo Vallina-Rodriguez, Serge Egelman, et al. ""Won't somebody think of the children?" examining COPPA compliance at scale". In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2018.3 (2018), pp. 63–83.

[164] David Rodriguez, Akshath Jain, Jose M Del Alamo, and Norman Sadeh. "Comparing Privacy Label Disclosures of Apps Published in both the App Store and Google Play Stores". In: *2023 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2023, pp. 150–157.

[165] SafeUM. *Privacy Policy*. https://web.archive.org/web/20230220213832/https://safeum.com/privacypolicy.html. (Accessed on 10/10/2023). 2023.

[166] 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)". In: *Proceedings on Privacy Enhancing Technologies* (2023).

[167] Nikita Samarin, Alex Sanchez, Trinity Chung, Akshay Dan Bhavish Juleemun, Conor Gilsenan, Nick Merrill, Joel Reardon, and Serge Egelman. "The Medium is the Message: How Secure Messaging Apps Leak Sensitive Data to Push Notification Services". In: *Proceedings on Privacy Enhancing Technologies* (2024).

[168] Madelyn R. Sanfilippo, Yan Shvartzshnaider, Irwin Reyes, Helen Nissenbaum, and Serge Egelman. "Disaster Privacy/Privacy Disaster". In: *Journal of the Association for Information Science and Technology* 71.9 (2020), pp. 1002–1014. DOI: https://doi.org/10.1002/asi.24353. eprint: https://asistdl.onlinelibrary.wiley.com/doi/pdf/10.1002/asi.24353. URL: https://asistdl.onlinelibrary.wiley.com/doi/abs/10.1002/asi.24353.

[169] Paul M Schwartz and Daniel J Solove. "The PII problem: Privacy and a new concept of personally identifiable information". In: *NYUL rev.* 86 (2011), p. 1814.

[170] Awanthika Senarath and Nalin AG Arachchilage. "Why developers cannot embed privacy into software systems? An empirical investigation". In: *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*. 2018, pp. 211–216.

[171] Awanthika Senarath, Marthie Grobler, and Nalin Asanka Gamagedara Arachchilage. "Will they use it or not? Investigating software developers' intention to follow privacy engineering methodologies". In: *ACM Transactions on Privacy and Security (TOPS)* 22.4 (2019), pp. 1–30.

[172] Jingyu Shi. *Notifying your users with FCM*. https://android-developers.googleblog.com/2018/09/notifying-your-users-with-fcm.html. (Accessed on 10/10/2023). 2023.

[173] Signal. *Grand jury subpoena for Signal user data, Central District of California (again!)* https://web.archive.org/web/20230921202338/https://signal.org/bigbrother/cd-california-grand-jury/. (Accessed on 10/10/2023). 2023.

[174] Signal. *Signal*. https://signal.org/. (Accessed on 10/10/2023). 2023.

[175] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D Breaux, and Jianwei Niu. "Toward a framework for detecting privacy policy violations in android application code". In: *Proceedings of the 38th International Conference on Software Engineering*. New York, NY, USA: ACM, 2016, pp. 25–36.

[176] Ivan Slobozhan, Tymofii Brik, and Rajesh Sharma. "Differentiable characteristics of Telegram mediums during protests in Belarus 2020". In: *Social Network Analysis and Mining* 13.1 (2023), p. 19.

[177] Adam Smith. *An Inquiry Into the Nature and Causes of the Wealth of Nations*. London, UK: Strahan and Cadell, 1776. ISBN: 9781546508649. URL: https://books.google.com/books?id=mt1SAAAAcAAJ.

[178] Daniel J Solove. "Against Privacy Essentialism". In: *GWU Legal Studies Research Paper Forthcoming* (2024).

[179] Daniel J Solove. "Introduction: Privacy self-management and the consent dilemma". In: *Harv. L. Rev.* 126 (2012), p. 1880.

[180] Daniel J Solove. *Understanding privacy*. Harvard university press, 2010.

[181] Sarah Spiekermann. "The challenges of privacy by design". In: *Communications of the ACM* 55.7 (2012), pp. 38–40.

[182] Sarah Spiekermann and Lorrie Faith Cranor. "Engineering privacy". In: *IEEE Transactions on software engineering* 35.1 (2008), pp. 67–82.

[183] Sarah Spiekermann and Jana Korunovska. "Towards a value theory for personal data". In: *Journal of Information Technology* 32.1 (2017), pp. 62–84.

[184] Sarah Spiekermann, Jana Korunovska, and Marc Langheinrich. "Inside the organization: Why privacy and security engineering is a challenge for engineers". In: *Proceedings of the IEEE* 107.3 (2018), pp. 600–615.

[185] statcounter. *Mobile Android Version Market Share United States Of America*. https://gs.statcounter.com/android-version-market-share/mobile/united-states-of-america/2021. 2021.

[186] StatCounter Global Stats. *Android Version Market Share Worldwide*. https://gs.statcounter.com/android-version-market-share/all/worldwide/2023. (Accessed on 06/01/2024). 2023.

[187] Anne Stopper and Jen Caltrider. *See no evil: Loopholes in Google's data safety labels keep companies in the clear and consumers in the dark. mozilla foundation*. Feb. 2023.

[188] Judith A Swanson. *The public and the private in Aristotle's political philosophy*. Cornell University Press, 1994.

[189] Mohammad Tahaei, Alisa Frik, and Kami Vaniea. "Privacy champions in software teams: Understanding their motivations, strategies, and challenges". In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 2021, pp. 1–15.

[190] Mohammad Tahaei, Kami Vaniea, and Awais Rashid. "Embedding privacy into design through software developers: Challenges and solutions". In: *IEEE Security & Privacy* 21.1 (2022), pp. 49–57.

[191] J. Tan, K. Nguyen, M. Theodorides, H. Negron-Arroyo, C. Thompson, S. Egelman, and D. Wagner. "The Effect of Developer-Specified Explanations for Permission Requests on Smartphone User Behavior". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 2014.

[192] Telegram. *Telegram Messenger*. https://telegram.org/. (Accessed on 10/10/2023). 2023.

[193] Telegram-FOSS on GitHub. *Notifications*. https://github.com/Telegram-FOSS-Team/Telegram-FOSS/blob/master/Notifications.md. (Accessed on 06/01/2024). 2024.

[194] The Drum. *WhatsApp's 3D billboard touts privacy features*. https://www.thedrum.com/news/2022/10/10/whatsapp-s-3d-billboard-touts-privacy-features. (Accessed on 10/10/2023). 2023.

[195] The Verge. *Now Mark Zuckerberg's making fun of Apple for iMessage, too*. https://www.theverge.com/2022/10/17/23409018/mark-zuckerberg-meta-whatsapp-imessage-privacy-security-ads. (Accessed on 10/10/2023). 2023.

[196] The White House. *National Cybersecurity Strategy*. https://www.whitehouse.gov/wp-content/uploads/2023/03/National-Cybersecurity-Strategy-2023.pdf. (Accessed on 06/01/2024). 2023.

[197] C. Thompson, M. Johnson, S. Egelman, D. Wagner, and J. King. "When It's Better to Ask Forgiveness than Get Permission: Designing Usable Audit Mechanisms for Mobile Permissions". In: *Proceedings of the 2013 Symposium on Usable Privacy and Security (SOUPS)*. 2013.

[198] Jan Tolsdorf, Michael Fischer, and Luigi Lo Iacono. "A case study on the implementation of the right of access in privacy dashboards". In: *Annual Privacy Forum*. Berlin, Germany: Springer, 2021, pp. 23–46.

[199] Horst Treiblmaier and Irene Pollach. "Users' perceptions of benefits and costs of personalization". In: *ICIS 2007 Proceedings* (2007), p. 141.

[200] Rahmadi Trimananda, Hieu Le, Hao Cui, Janice Tran Ho, Anastasia Shuba, and Athina Markopoulou. "{OVRseen}: Auditing Network Traffic and Privacy Policies in Oculus {VR}". In: *31st USENIX security symposium (USENIX security 22)*. Berkeley, CA, USA: USENIX, 2022, pp. 3789–3806.

[201] One Trust. *One Trust*. https://www.onetrust.com/. Accessed: 2022-08-31. 2021.

[202] L. Tsai, P. Wijesekera, J. Reardon, I. Reyes, S. Egelman, D. Wagner, N. Good, and J. Chen. "Turtle Guard: Helping Android Users Apply Contextual Privacy Preferences". In: *Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017)*. Santa Clara, CA: USENIX Association, 2017, pp. 145–162. ISBN: 978-1-931971-39-3. URL: https://www.usenix.org/conference/soups2017/technical-sessions/presentation/tsai.

[203] Zeynep Tufekci. *Twitter and tear gas: The power and fragility of networked protest*. Yale University Press, 2017.

[204] U.S. Congress. *H.R.4952 - Electronic Communications Privacy Act of 1986*. https://www.congress.gov/bill/99th-congress/house-bill/4952. (Accessed on 10/10/2023). 1986.

[205] UnifiedPush. *UnifiedPush*. https://unifiedpush.org/. (Accessed on 10/10/2023). 2023.

[206] United States District Court for the Central District of California. *Application for a Warrant re: Case No. 2:22-MJ-03119*. https://www.documentcloud.org/documents/24192891-search-warrant-for-google-account-for-push-notification-data. (Accessed on 06/01/2024). 2022.

[207] United States District Court for the District of Columbia. *Application for a Warrant re: Case No. 21-sc-270*. https://www.documentcloud.org/documents/24192911-6d68977d-f8ef-4080-9742-290cff8a6c28. (Accessed on 06/01/2024). 2021.

[208] Tobias Urban, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. ""Your hashed IP address: Ubuntu." perspectives on transparency tools for online advertising". In: *Proceedings of the 35th Annual Computer Security Applications Conference*. New York, NY, USA: ACM, 2019, pp. 702–717.

[209] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. "A study on subject data access in online advertising after the GDPR". In: *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Berlin, Germany: Springer, 2019, pp. 61–79.

[210] Tobias Urban, Dennis Tatang, Martin Degeling, Thorsten Holz, and Norbert Pohlmann. "The unwanted sharing economy: An analysis of cookie syncing and user transparency under GDPR". In: *arXiv preprint arXiv:1811.08660* (2018), pp. 1–26.

[211] Aleksandra Urman, Justin Chun-ting Ho, and Stefan Katz. "Analyzing protest mobilization on Telegram: The case of 2019 anti-extradition bill movement in Hong Kong". In: *Plos one* 16.10 (2021), e0256675.

[212] Maggie Van Nortwick and Christo Wilson. "Setting the Bar Low: Are Websites Complying With the Minimum Requirements of the CCPA?" In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2022.1 (2022), pp. 608–628.

[213] Sophie Veys, Daniel Serrano, Madison Stamos, Margot Herman, Nathan Reitinger, Michelle L Mazurek, and Blase Ur. "Pursuing Usable and Useful Data Downloads Under GDPR/CCPA Access Rights via Co-Design". In: *Seventeenth Symposium on Usable Privacy and Security (SOUPS 2021)*. Berkeley, CA, USA: USENIX, 2021, pp. 217–242.

[214] Viber. *Privacy Notice for California Residents*. https://web.archive.org/web/20230310001732/https://www.viber.com/en/terms/ccpa-privacy-rights/. (Accessed on 10/10/2023). 2023.

[215] California Voters. *California Privacy Rights Act of 2020 (CPRA)*. Proposition 24, amending the California Consumer Privacy Act, codified in California Civil Code §§ 1798.100–1798.199.100. Approved November 3, 2020; most provisions effective January 1, 2023. 2020. URL: https://leginfo.legislature.ca.gov/faces/codes_displayText.xhtml?division=3.&part=4.&lawCode=CIV&title=1.81.5.

[216] Xiaoyin Wang, Xue Qin, Mitra Bokaei Hosseini, Rocky Slavin, Travis D Breaux, and Jianwei Niu. "Guileak: Tracing privacy policy claims on user input data for android applications". In: *Proceedings of the 40th International Conference on Software Engineering*. New York, NY, USA: ACM, 2018, pp. 37–47.

[217] Ian Warren, Andrew Meads, Satish Srirama, Thiranjith Weerasinghe, and Carlos Paniagua. "Push notification mechanisms for pervasive smartphone applications". In: *IEEE Pervasive Computing* 13.2 (2014), pp. 61–71.

[218] Samuel Warren and Louis Brandeis. "The right to privacy". In: *Killing the Messenger: 100 Years of Media Criticism*. Columbia University Press, 1989, pp. 1–21.

[219] Miranda Wei, Madison Stamos, Sophie Veys, Nathan Reitinger, Justin Goodman, Margot Herman, Dorota Filipczuk, Ben Weinshel, Michelle L Mazurek, and Blase Ur. "What Twitter knows: Characterizing ad targeting practices, user perceptions, and ad explanations through users' own Twitter data". In: *29th USENIX Security Symposium (USENIX Security 20)*. Berkeley, CA, USA: USENIX, 2020, pp. 145–162.

[220] Alan F Westin. "Privacy and freedom". In: *Washington and Lee Law Review* 25.1 (1968), p. 166.

[221] Mark Wickham. "Push Messaging". In: *Practical Android: 14 Complete Projects on Advanced Techniques and Approaches* (2018), pp. 135–172.

[222] Primal Wijesekera, Arjun Baokar, Ashkan Hosseini, Serge Egelman, David Wagner, and Konstantin Beznosov. "Android permissions remystified: A field study on contextual integrity". In: *24th USENIX Security Symposium (USENIX Security 15)*. 2015, pp. 499–514.

[223] Primal Wijesekera, Arjun Baokar, Lynn Tsai, Joel Reardon, Serge Egelman, David Wagner, and Konstantin Beznosov. "The feasibility of dynamically granted permissions: Aligning mobile privacy with user preferences". In: *2017 IEEE Symposium on Security and Privacy (SP)*. New York, NY, USA: IEEE, 2017, pp. 1077–1093.

[224] Primal Wijesekera, Joel Reardon, Irwin Reyes, Lynn Tsai, Jung-Wei Chen, Nathan Good, David Wagner, Konstantin Beznosov, and Serge Egelman. "Contextualizing privacy decisions for better prediction (and protection)". In: *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 2018, pp. 1–13.

[225] Wikipedia. *Chicken Gun*. https://en.wikipedia.org/wiki/Chicken_gun. 2023.

[226] Janis Wong and Tristan Henderson. "The right to data portability in practice: exploring the implications of the technologically neutral GDPR". In: *International Data Privacy Law* 9.3 (2019), pp. 173–191.

[227] Kim Wuyts, Laurens Sion, and Wouter Joosen. "Linddun go: A lightweight approach to privacy threat modeling". In: *2020 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. IEEE. 2020, pp. 302–309.

[228] Ron Wyden. *Wyden Smartphone Push Notification Surveillance Letter*. https://www.wyden.senate.gov/imo/media/doc/wyden_smartphone_push_notification_surveillance_letter.pdf. (Accessed on 01/01/2024). Dec. 2023.

[229] 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 at scale". In: *arXiv preprint arXiv:2206.06274* (2022).

[230] Zhi Xu and Sencun Zhu. "Abusing Notification Services on Smartphones for Phishing and Spamming." In: *WOOT*. 2012, pp. 1–11.

[231] Jinyan Zang, Krysta Dummit, James Graves, Paul Lisker, and Latanya Sweeney. "Who knows what about me? A survey of behind the scenes personal data sharing to third parties by mobile apps". In: *Technology Science* 30 (2015), pp. 1–53.

[232] Sebastian Zimmeck, Rafael Goldstein, and David Baraka. "PrivacyFlash Pro: Automating Privacy Policy Generation for Mobile Apps." In: *NDSS*. Reston, VA, USA: Internet Society, 2021.

[233] Sebastian Zimmeck, Peter Story, Daniel Smullen, Abhilasha Ravichander, Ziqi Wang, Joel R Reidenberg, N Cameron Russell, and Norman Sadeh. "MAPS: Scaling privacy compliance analysis to a million apps". In: *Proceedings on Privacy Enhancing Technologies (PoPETs)* 2019.3 (2019), pp. 66–86.

[234] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman M Sadeh, Steven M Bellovin, and Joel R Reidenberg. "Automated Analysis of Privacy Requirements for Mobile Apps." In: *NDSS*. Reston, VA, USA: Internet Society, 2017.

[235] Yixin Zou, Abraham H Mhaidli, Austin McCall, and Florian Schaub. ""I've Got Nothing to Lose": Consumers' Risk Perceptions and Protective Actions after the Equifax Data Breach". In: *Fourteenth Symposium on Usable Privacy and Security (SOUPS 2018)*. 2018, pp. 197–216.

[236] Shoshana Zuboff. "The age of surveillance capitalism". In: *Social theory re-wired*. Routledge, 2023, pp. 203–213.

Appendix A

Supplemental Materials for Chapter 4

A.1 VCR email templates

This appendix contains the email templates that we used to submit verifiable consumer requests to app developers, as well as the conditions under which we sent it. Note that we cited the provision *Cal. Civil Code 1798.140* in the template emails to direct the developers to the list of categories predefined by the CCPA to facilitate their response and to improve the consistency of categorization across different companies.

Initial Request

We used the following email template to initiate the VCR:

Dear Privacy Compliance Officer,

My name is [*name*]. I live in California and I am exercising my data access rights under the California Consumer Privacy Act (CCPA) to obtain a copy of the categories and the specific pieces of personal information that [*company*] has collected about me.

I'm requesting a copy of any and all of the records you have pertaining to me including (but not limited to):

1. Specific pieces of personal information and any persistent identifiers that you have collected about me including all information or content provided or posted by me, any information you have collected about me, or any personal information you have obtained or acquired about me from a third party business or service provider;
2. Categories of personal information you have collected about me pursuant to the enumerated list of categories in Cal. Civil Code 1798.140(o);
3. Categories of sources from which my personal information is collected;
4. Categories of personal information that you have sold or disclosed for a business purpose

about me by each category of personal information enumerated in Cal. Civil Code 1798.140(o);

5. Third parties to whom my personal information was sold or disclosed for a business purpose; and
6. The business or commercial purpose for collecting or selling my personal information.

I expect a confirmation of receipt within 10 business days and information about how [*company*] will process my request, sent to this email address. Please let me know if you need any more information from me as soon as possible.

If you believe that you are not subject to the CCPA, please reply back as soon as possible and let me know why you believe the CCPA does not apply in this case.

Sincerely,

[*Name*]

Unable to perform request

Company has directed us to use an alternative method to submit VCR that does not provide access to the full records.

Dear [*Name of Privacy Compliance Officer*],

Thank you for your reply. Unfortunately, the [*alternative request method*] that you have directed me to use to submit my request does not allow me to fully exercise my data access rights under the California Consumer Privacy Act.

Specifically, the [*alternative request method*] does not allow me to request a copy of the following records you have pertaining to me:

(Select and include the appropriate ones in the email)

1. Specific pieces of personal information and any persistent identifiers that you have collected about me including all information or content provided or posted by me, any information you have collected about me, or any personal information you have obtained or acquired about me from a third party business or service provider;
2. Categories of personal information you have collected about me pursuant to the enumerated list of categories in Cal. Civil Code 1798.140(o);
3. Categories of sources from which my personal information is collected;
4. Categories of personal information that you have sold or disclosed for a business purpose about me by each category of personal information enumerated in Cal. Civil Code 1798.140(o);
5. Third parties to whom my personal information was sold or disclosed for a business purpose; and
6. The business or commercial purpose for collecting or selling my personal information.

Please let me know how I should proceed as soon as possible.

Sincerely,

[*Name*]

Missing Information Request

Company responded to our VCR without providing all of the requested information.

Dear [*Name of Privacy Compliance Officer*],

Thank you for your reply. Unfortunately, the copy of the records that I have received does not contain all of the requested information. Specifically, I have not received a copy of the following records you have pertaining to me:

(Select and include the appropriate ones in the email)

1. Specific pieces of personal information and any persistent identifiers that you have collected about me including all information or content provided or posted by me, any information you have collected about me, or any personal information you have obtained or acquired about me from a third party business or service provider;
2. Categories of personal information you have collected about me pursuant to the enumerated list of categories in Cal. Civil Code 1798.140(o);
3. Categories of sources from which my personal information is collected;
4. Categories of personal information that you have sold or disclosed for a business purpose about me by each category of personal information enumerated in Cal. Civil Code 1798.140(o);
5. Third parties to whom my personal information was sold or disclosed for a business purpose; and
6. The business or commercial purpose for collecting or selling my personal information.

Please let me know how I should proceed as soon as possible.

Sincerely,

[*Name*]

Account Holder Verification Request

We created an account with the app and the developer required us to furnish documentation to verify our identity that we could not provide.

Dear [*Name of Privacy Compliance Officer*],

Thank you for your reply. Unfortunately, I prefer not to provide the information that you have requested to verify my identity, as I believe it to be invasive and beyond the requirements of the CCPA.

As an account holder with [*company*], I would prefer verifying my identity using existing authentication practices for my account per CCR § 999.324(a). For your convenience, the [*email address OR username*] associated with my account is [*email address OR username*].

Please let me know if you need any more information from me as soon as possible.

Sincerely,

[*Name*]

Account Non-Holder Verification Request

We *did not* create an account with the app and the developer required us to furnish documentation to verify our identity that we could not provide.

Dear [*Name of Privacy Compliance Officer*],

Thank you for your reply. Unfortunately, I prefer not to provide the information that you have requested to verify my identity, as I believe it to be invasive and beyond the requirements of the CCPA.

Instead, I would prefer verifying my identity by matching the following three pieces of personally identifiable information that I have previously provided to [*company*] per CCR § 999.325(b) and (c):

(Select and include the appropriate ones in the email)

1. *PII1 Type: PII1 Value*
2. *PII2 Type: PII2 Value*
3. *PII3 Type: PII3 Value*

Please let me know if you need any more information from me as soon as possible.

Sincerely,

[*Name*]

First Follow-Up

Company did not respond to our initial request in 10 business days.

Dear Privacy Compliance Officer,

My name is [*name*] and I am following up on a request I made on [*date*] to access the personal information that [*company*] has collected about me. I was expecting to receive a confirmation of receipt and information about how [*company*] would process my request within 10 business days per 11 CCR § 999.313(a). For your convenience, my original request is as follows:

I'm requesting a copy of any and all of the records you have pertaining to me including (but not limited to):

1. Specific pieces of personal information and any persistent identifiers that you have collected about me including all information or content provided or posted by me, any information you have collected about me, or any personal information you have obtained or acquired about me from a third party business or service provider;
2. Categories of personal information you have collected about me pursuant to the enumerated list of categories in Cal. Civil Code 1798.140(o);
3. Categories of sources from which my personal information is collected;
4. Categories of personal information that you have sold or disclosed for a business purpose about me by each category of personal information enumerated in Cal. Civil Code 1798.140(o);
5. Third parties to whom my personal information was sold or disclosed for a business purpose; and
6. The business or commercial purpose for collecting or selling my personal information.

I expect a reply to this email address as soon as possible. If you believe that you are not subject to the California Consumer Privacy Act (CCPA), please reply back as soon as possible and let me know why you believe the CCPA does not apply in this case.

Sincerely,

[*Name*]

Second Follow-Up

Company did not respond to our first follow-up email in 10 business days.

Dear Privacy Compliance Officer,

My name is [*name*] and I am following up on a request I originally made on [*date*] to access the personal information that [*company*] has collected about me. I have previously followed up about my request on [*date*], but I have not heard back from you. I was expecting to receive a confirmation of receipt and information about how [*company*] would process my request within 10 business days per 11 CCR § 999.313(a). My original request is as follows:
I'm requesting a copy of any and all of the records you have pertaining to me including (but

not limited to):

1. Specific pieces of personal information and any persistent identifiers that you have collected about me including all information or content provided or posted by me, any information you have collected about me, or any personal information you have obtained or acquired about me from a third party business or service provider;
2. Categories of personal information you have collected about me pursuant to the enumerated list of categories in Cal. Civil Code 1798.140(o);
3. Categories of sources from which my personal information is collected;
4. Categories of personal information that you have sold or disclosed for a business purpose about me by each category of personal information enumerated in Cal. Civil Code 1798.140(o);
5. Third parties to whom my personal information was sold or disclosed for a business purpose; and
6. The business or commercial purpose for collecting or selling my personal information.

I expect a reply to this email address as soon as possible. If you believe that you are not subject to the California Consumer Privacy Act (CCPA), please reply back as soon as possible and let me know why you believe the CCPA does not apply in this case.

Sincerely,

[*Name*]

A.2 Codebook

Tables A.1 and A.2 include the codebook that we used to perform a qualitative analysis of disclosures in privacy policies. We use the categories of personal information defined in Cal. Civil Code 1798.140 to represent the codes for the collection and sharing in Table A.1. Table A.2 contains our codes for the categories of third parties.

For each privacy policy, coders saw the following prompts:

- Does this app developer include disclosures that reference the CCPA, either as part of the general privacy policy or as a standalone document?
- Does the privacy policy state that the app developer collects [*PII Code*]?
- Does the privacy policy state that the app developer discloses or shares [*PII Code*]?
- Does the privacy policy state that the app developer shares personal information with [*Third Party Code*]?

A.3 Data taxonomy

Table A.3 enumerates the 7 categories of personal information defined in the CCPA relevant to this work, our subcategories, as well as the types and values of personal information that we have predefined for each test device.

We generated pseudonymous data for *User Identifiers*, *Customer Records*, *Protected Classifications*, *Professional* and *Education Information* using publicly-available random value generators, such as those found on the Random Lists[1] website and the Faker[2] Python package. We obtained other types of personal information, including *Device Identifiers* and *Geolocation Data*, directly from our test devices.

[1] https://www.randomlists.com/
[2] https://pypi.org/project/Faker/0.7.4/

Category	Description
Identifiers	Real name, alias, postal address, unique personal identifier, online identifier, IP address, email address, account name, social security number, driver's license number, passport number, or other similar identifiers.
Customer Records	Name, signature, social security number, physical characteristics or description, address, telephone number, passport number, driver's license or state identification card number, insurance policy number, education, employment, employment history, bank account number, credit card number, debit card number, or any other financial information, medical or health insurance information.
Characteristics of **Protected Classifications** under California or Federal Law	Age, race, color, ancestry, national origin, citizenship, religion or creed, marital status, medical condition, physical or mental disability, sex, gender, gender identity, gender expression, pregnancy or childbirth and related medical conditions, sexual orientation, veteran or military status, genetic information (including familial genetic information).
Commercial Information	Records of personal property, products or services purchased, obtained, or considered, or other purchasing or consuming histories or tendencies.
Biometric Information	Genetic, physiological, behavioral, and biological characteristics, or activity patterns used to extract a template or other identifier or identifying information, such as, fingerprints, faceprints, and voiceprints, iris or retina scans, keystroke, gait, or other physical patterns, sleep, health, or exercise data.
Network Activity	Browsing history, search history, or information regarding a consumer's interaction with a website, application, or advertisement.
Geolocation Data	Information such as physical location or movements.
Sensory Data	Audio, electronic, visual, thermal, olfactory, or similar information.
Professional Information	Information such as current or past job history or performance evaluations.
Education Information	Education records directly related to a student maintained by an educational institution or party acting on its behalf, such as grades, transcripts, class lists, student schedules, student identification codes, student financial information, or student disciplinary records.
Inferences	Consumer's preferences, characteristics, psychological trends, predispositions, behavior, attitudes, intelligence, abilities, or aptitudes.

Table A.1: Categories of personal information.

Category	Description
Affiliated Companies	Companies related to the app developer through ownership, such as when the app developer holds a stake in the company (e.g., a subsidiary) or when another third party controls both the company and the app developer.
Advertising Networks	Connect advertisers to websites or apps (the "publishers") that want to host advertisements.
Marketing Providers	Offer products, services, or other promotions to the app's users, for instance, by calling, texting or emailing them with marketing messages.
Analytics Providers	Capture data about the app's audience in order to identify unique users, track their interactions, and record their behavior for the purpose of improving the app, informing company strategy, or general research.
Security and Fraud	Provide tools, such as identity verification and fraud detection, to prevent fraudulent activity, improve app security, enforce terms of service, and protect users and property.
Payment Processors	Enable merchants to sell products and accept in-app card payments.
Customer Support	Provide tools to collect, organize, respond to, and report on customer support requests tounderstand user needs, provide assistance, and streamline communication.
Storage and Infrastructure	Provide services, such as data hosting, cloud storage, load balancing and other infrastructure to optimize content delivery and performance.
Search Engines	Collect, organize and enable the search for content online, including information generated by users interacting with the app or other users.
Social Media Platforms	Provide technologies and means of communication, through which users create and share information and ideas in online communities.
Order Fulfillment	Process orders and deliver products to customers.
Law Enforcement	Sharing to comply with a legal obligation or a request from regulators, courts, law enforcement, and other governmental agencies.
Unspecified Partners	Sharing with unspecified partners and service providers.

Table A.2: Categories of data recipients.

CCPA Category	Subcategory	Description	PII Types	Example Values
Identifiers	User	Identifiers set by the user	Usernames, email address, website	schneider90christopher19
	Network	Identifiers unique to user's network	IP Address, router MAC and SSID	135.***.***.79, 48:**:**:**:**:06
	Device	Identifiers unique to user's device	Android advertising ID (AAID), hardware IDs, IMEI, IMSI, SIM ID, Wi-Fi MAC, GSFID	97PAY11GN2, 359677097304580, 58:CB:52:8B:C8:66, 03140e43-9bb7-[...]
	App	Identifiers unique to a single app	Android ID, app fingerprint ID, identity ID	7892f8834ddbf2df 1039977256339324001
Customer Records	Customer	Information about the user	Name, phone number, height, weight	Christopher Schneider, 323-448-****
	Contacts	Information about user's contacts	Contact name, contact phone number	Scott Pratt, 415-200-****
	Residence	Information about user's general address of residence	Street, city, county, ZIP Code	957 Green Causeway, Los Angeles
Protected Classifications	—	Information protected under the California and U.S. federal laws	Gender, date of birth, age	Male, 20-May-1990
Geolocation Data	Precise	Locates a specific building	Precise longitude/latitude coordinates, street name	*****
	Coarse	Does not locate a specific building	Coarse longitude/latitude coordinates, city, county, ZIP Code	*****
Sensory Data	—	Audio, electronic, visual, thermal, olfactory, or similar information	Accelerometer, gyroscope, magnetometer readings	AK0991X, BMI160
Professional Information	—	Current or past job history or performance evaluations	Job, company	Clinical Psychologist, Williams and Davis
Education Information	—	Education records directly related to a student	College	Villanova University

Some of the values have been redacted to preserve the privacy of researchers to whom the data pertains.

Table A.3: Types of personal information that we used during our app measurements.

Appendix B

Supplemental Materials for Chapter 5

B.1 Data Types

Table B.1 enumerates the data types that we searched for during our analysis of Android apps. Google defines and uses these data types to populate the information presented to users in the form of privacy labels in the app's listing on Google Play Store [83].

B.2 Code Analysis Workflow

We used this set of questions to analyze the source code of apps in our data set. These questions can also assist with data flow mapping, or in other words, tracing data contained

Data Type	Description
Device or other IDs	Identifiers that relate to an individual device, browser or app. For example, an IMEI number, MAC address, Widevine Device ID, Firebase installation ID, or advertising identifier.
User IDs	Identifiers that relate to an identifiable person. For example, an account ID, account number, or account name.
Name	How a user refers to themselves, such as their first or last name, or nickname.
Phone number	A user's phone number.
Messages	Any other types of messages. For example, instant messages or chat content.

Table B.1: Google Play Store's data types applicable to our study.

in a push notification from its creation until the notification is displayed to the user.

- Does the app's `AndroidManifest.xml` register a service that `extends FirebaseMessagingService`?
- Locate the Java .java (or Kotlin .kt) source file corresponding to the registered service.
- Which FCM methods (e.g., `onMessageReceived()`, `onNewToken()`, etc.) does the service override?
- The `onMessageReceived()` method gets invoked when the client app receives an FCM push notification. Does the service override `onMessageReceived()` method?
- Data payload contained in an FCM push notification can be accessed by calling `remoteMessage.getData()`. Does the `onMessageReceived()` method invoke `getData()` on its argument of type `RemoteMessage`?
- Is there any indication that `RemoteMessage` contains sensitive data, based on the names of the keys or logging?
- Trace the code execution from the `onMessageReceived()` method until the message is displayed to the user.
- Does `RemoteMessage` get passed as a parameter to any function?
- What mechanisms (if any) are in place to ensure that notification contents do not get leaked to Google's FCM server?